

Vol. 2 No. 2  
April, 1993

**\$4.95**

# ATARI CLASSICS

**The Magazine for the Dedicated 8-Bit User**

*In This Issue:*

Advanced C

Programming, Part 2

Us Guys Iz Yooze Guys

Type 'n' Talk For Cheep Talk

A Look At Synchromesh

Text Adventures From Level 9

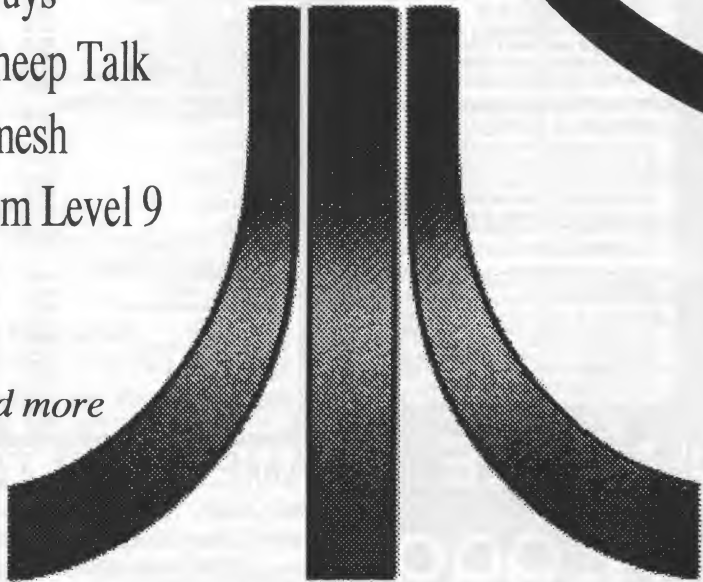
News Roundup

Tips 'n' Tricks

*...and more*



Ram Upgrades:  
2 Articles!



**AC Is Growing! Subscribe Now!**

# ATARI 8-BIT

## SOFTWARE - PARTS AND BOOKS FOR ATARI XL / XE & 800/400 COMPUTERS

### 1050 MECHANISM

Factory fresh TANDON mechs. make difficult repairs a snap. Units are complete with Head, Stepper, Spindle motor, belt etc. Just plug in, no difficult alignments or adjustments required.

\*Special\* \$39<sup>95</sup>  
2 for \$70.00

### POWER PACKS

Exact replacement transformer for 800/400, 1050 810, 1200XL, 850, and XF551 units. Part #CO7945. \$14<sup>50</sup> ea.

### XL/XE SUPPLY

Power Pak for 800XL, 600XL, 130XE, 65XE & XE Game. \$25<sup>00</sup>

### KEYBOARDS

New and complete subassembly. Easy internal replacement.

130XE/65XE \$35.00  
800 ..... \$40.00  
800XL ..... \$29.50

### XL 40 PIN LSI CHIP SET

A Complete set of 40 Pin Large Scale Integrated Circuits for your 800XL, 600XL or 1200XL computer. Great for quick repairs! Set contains one each of the following: CPU, GTIA, ANTIC, PIA AND POKEY. \$14<sup>95</sup>

### 800 5 PIECE BOARD SET

Includes Main Board, Power Board CPU, 10K O.S. and 16K RAM Board. All boards new, tested and fully populated. Set includes instructions that will allow you to repair ANY 800 Computer \$28.50

### 1050 REPAIRS

We are now offering a limited repair service on the Atari 1050 Disk Drive. Please call our Service Dept. at 510-352-3787 for a Repair Authorization Number. Units without RA# will not be accepted. Service rate... \$69.95

### MISCELLANEOUS

600XL 64K UPGRADE KIT ..... \$19.95  
13 PIN SERIAL I/O CABLE ..... \$4.50  
SPECIAL - 3 I/O CABLES FOR \$10.00  
ACE JOYSTICK ..... \$7.95  
POWERPLAYER JOYSTICK ..... \$12.95  
COMPUTER / TV SWITCH BOX ..... \$6.95  
400 BOARD SET (W/O Kbd) ..... \$18.95  
PADDLE CONTROLLERS ..... \$9.95  
REV. "C" BASIC ROM IC ..... \$15.00  
Newell 256K 800XL Upgrade ..... \$27.50  
Set of 256K RAM CHIPS ..... \$19.95  
800XL 1 Meg Upgrade W/O Ram ..... \$42.50  
130XE 1 Meg Upgrade W/O Ram ..... \$39.95  
850 or PR PRINTER CABLE ..... \$14.50  
8-BIT MONITOR CABLE ..... \$9.95  
XEGS Keyboard Extension Cable ..... \$14.50  
I/O 13 PIN PC CONNECTOR ..... \$4.50  
I/O 13 PIN PLUG KIT (Cable End) ..... \$4.50  
RF CABLE (COMPUTER TO TV) ..... \$3.75  
1050 TRACK ZERO SENSOR ..... \$8.50  
Animation Station Graphics Tablet ..... \$64.95  
PRINTER INTERFACE ..... \$43.95

## BOOKS

Mapping the Atari (XL/XE) ..... \$24.95  
Mapping the Atari (800/400) ..... \$12.95  
Compute's First Book of Atari ..... \$9.95  
Compute's Third Book of Atari  
(Includes a high quality type in Word Processor) ..... \$9.95  
First & Third Book Combo ..... \$14.95  
First Book of Atari Graphics ..... \$9.95  
2nd Book of Atari Graphics ..... \$9.95  
Graphic Book Combo #1 & 2 ..... \$14.95  
XE USERS HANDBOOK ..... \$24.95  
XL USERS HANDBOOK ..... \$24.95  
Assembly Language Guide ..... \$24.95  
USING NEWSROOM ..... \$19.95  
Programmers Ref. Guide ..... \$9.95  
The Creative PrintMaster ..... \$7.95  
Write Your Own Games ..... \$5.00  
How to 6502 Program ..... \$5.00  
Basic Reference Manual ..... \$5.00  
INSIDE ATARI BASIC ..... \$5.00  
DOS 2.0 Reference Manual ..... \$7.50  
Tech Ref. Notes for 800/400 ..... \$19.95  
The Computer Playground ..... \$5.95  
SpeedScript Word Processor ..... \$14.95

### 1st XLENT Word Processor

A full-featured, word processor that is easy to learn, ICON based and can be used with a joystick. Edit up to two documents at once. Preview mode shows your document in simulated 80 columns, just as it will appear on your printer. DISK. \$25.00

## PRINTER INTERFACE

MICROPRINT INTERFACE  
ALLOWS YOU TO CONNECT ANY  
STANDARD PARALLEL PRINTER  
TO YOUR ATARI 8-BIT. WORKS  
WITH 800XL, 130XE, 800/400,  
65XE, XEGS, 600XL. (1200XL  
requires easy modification). \$43.95

### BOOKKEEPER & KEYPAD

You get both Atari's 8 bit professional bookkeeping system and the handy CX85 numeric keypad for one low price. Packages Factory sealed 4 DISK SET \$19<sup>95</sup>

### VISCALC SPREADSHEET

Unleash the computing power of your 8 bit Atari with Viscalc. Compute everything from home finances to high powered financial projections. Hundreds of uses DISK \$19<sup>95</sup>

### Super DataBase 1-2-3

INNOVATIVE PRODUCT CREATES AMAZING DATABASE APPLICATIONS EVEN IF YOU HAVE NO PROGRAMMING SKILL. ONCE YOU CREATE A PROGRAM WITH SUPER DATABASE IT BECOMES A STAND-ALONE APPLICATION THAT RUNS ON ITS OWN. "GENERATE" YOUR OWN PROGRAMS TO KEEP TRACK OF EVERYTHING. INCLUDES 137 PAGE MANUAL - 3 DISKS \$59<sup>95</sup>

### CARTRIDGES FOR XL/XE 800/400

ATARIWRITER Word Processor Cart. \$29.95  
ATARI LOGO Cartridge ..... \$14.95  
POLE POSITION Cartridge ..... \$12.50  
3-D TIC TAC TOE Cartridge ..... \$9.95  
CHICKEN Cartridge ..... \$9.95  
JOUST Cartridge ..... \$9.95  
HARDBALL Cartridge ..... \$9.95  
DARK CHAMBERS Cartridge ..... \$14.50  
THUNDERFOX Cartridge ..... \$14.50  
DESERT FALCON Cartridge ..... \$14.50  
LODERUNNER Cartridge ..... \$14.50  
JUNGLE HUNT Cartridge ..... \$9.95  
STAR TREK Cart. by SEGA ..... \$9.95  
PAC-MAN Cartridge ..... \$4.00  
DONKEY KONG Cartridge ..... \$5.00  
DONKEY KONG JR. Cartridge ..... \$9.95  
DELUXE INVADERS Cartridge ..... \$4.00  
STAR RAIDERS Cartridge ..... \$5.00  
WIZARD OF WOR Cartridge ..... \$9.95  
CROSSBOW Cart. Req. Lightgun ..... \$14.50  
FOODFIGHT Cart. XL/XE Only. ..... \$9.95  
BLUE MAX Cartridge ..... \$9.95  
MOON PATROL Cartridge ..... \$9.95  
ARCHON Cartridge ..... \$12.50  
SLIME Cartridge ..... \$7.50  
E. T. PHONE HOME Cartridge ..... \$9.95  
ACE OF ACES Cartridge ..... \$14.50  
BATTLEZONE Cartridge ..... \$12.50  
CAVERNS OF MARS Cartridge ..... \$9.95  
GATO Cartridge ..... \$12.50  
SUPER BREAKOUT Cartridge ..... \$12.50  
RESCUE ON FRONTALAS Cart. ..... \$12.50  
ONE ON ONE BASKETBALL Cart. ..... \$17.50  
NECROMANCER Cartridge ..... \$9.95  
MILLPEDE Cartridge ..... \$9.95  
FINAL LEGACY Cartridge ..... \$9.95  
PLATTER MANIA Cart. by Epyx ..... \$5.00  
PENGU Cartridge ..... \$9.95  
EASTERN FRONT Cartridge ..... \$12.50  
ADVENTURE CREATOR Cartridge ..... \$12.50  
SILICON WARRIOR Cartridge ..... \$9.95  
MICROSOFT BASIC II Cartridge ..... \$26.95  
BASIC LANGUAGE Cartridge ..... \$15.00  
Editor/Assembler Cart. (No Manual) ..... \$9.95  
PILOT LANGUAGE PACKAGE ..... \$12.50  
MATH ENCOUNTERS ..... \$12.50

### SPINNAKER EDUCATIONAL CARTRIDGES

FRACTION FEVER Cartridge ..... \$8.75  
FACEMAKER Cartridge ..... \$8.75  
ALPHABET ZOO Cartridge ..... \$8.75  
UP FOR GRABS Cartridge ..... \$8.75  
ALF IN THE COLOR CAVES Cart. ..... \$8.75  
DELTA DRAWING Cartridge ..... \$8.75  
STORY MACHINE 800/400 only ..... \$8.75  
LINKING LOGIC (Fisher-Price) Cart. ..... \$8.75  
GRANDMA'S HOUSE (Disk) ..... \$8.75

### DISKS XL/XE - 800

VIDEO TITLE SHOP 3 DISK BONUS PACK  
Contains VTS - 2 Companion Disks \$14.95  
TARGET PRACTICE ..... \$12.50  
ROSEN'S BRIGADE ..... \$12.50  
SEA BANDIT ..... \$12.50  
SPIDER QUAKE ..... \$12.50  
STARBASE FIGHTER ..... \$12.50  
MANIAC MINER ..... \$12.50  
ALTERNATE REALITY (The City) ..... \$12.50  
ALTERNATE REALITY The Dungeon ..... \$12.50  
SARACEN Disk ..... \$9.95  
GUNSLINGER Disk (64K) ..... \$9.95  
TOMAHAWK Helicopter Game 64K ..... \$12.50  
THEATER EUROPE War Game ..... \$12.50  
BRUCE LEE Disk ..... \$12.50  
MERCENARY Disk ..... \$12.50  
CONAN Disk ..... \$12.50  
221-B BAKER STREET (64K) ..... \$12.50  
Napoleon in Russia (Borodino 1812) ..... \$9.95  
BISMARCK ..... \$9.95  
ZORRO ..... \$9.95  
MOONSHUTTLE ..... \$4.95  
CROSSCHECK ..... \$7.50

### MICROPROSE SIMULATIONS

F-15 STRIKE EAGLE ..... \$8.95  
TOP GUNNER COLLECTION ..... \$8.95  
SILENT SERVICE ..... \$8.95  
KENNEDY APPROACH ..... \$8.95  
DECISION IN THE DESERT ..... \$8.95  
CRUSADE IN EUROPE ..... \$8.95  
CONFLICT IN VIETNAM ..... \$8.95  
HELLCAT ACE ..... \$8.95

### INFOMCOM ADVENTURES

WISHBRINGER ..... \$9.95  
MOONMIST ..... \$9.95  
SEA STALKER ..... \$9.95  
INFIDEL ..... \$9.95  
CUT THROATS ..... \$9.95  
BALLYHOO ..... \$9.95  
SUSPENDED ..... \$9.95  
ZORK 1 ..... \$9.95  
PLANETFALL ..... \$14.95  
HOLLYWOOD HIJINX ..... \$14.95

### Disks XL's/XE's/800

CYCLE KNIGHT ..... \$9.95  
GOLDEN OLDIES ..... \$9.95  
SUMMER GAMES BY EPYX ..... \$4.75  
SPIDERMAN ..... \$4.95  
REAR GUARD ..... \$4.95  
ROCKET REPAIRMAN ..... \$4.95  
SPEED KING ..... \$4.95  
STRATOS ..... \$4.95  
L.A. SWAT ..... \$7.50  
THE GAMBLER ..... \$7.50  
CROSS COUNTRY ROAD RACE ..... \$9.95

### Disks XL's/XE's/800

MULE ..... \$29.95  
ULTIMA III (EXODUS) ..... \$29.95  
ChromaCad2 Model Builder/Shader ..... \$29.95  
1ST XLENT WORD PROCESSOR ..... \$25.00  
BLOCKABOO (GREAT NEW GAME) ..... \$14.95  
AUTO DUEL ..... \$15.00  
GAUNTLET (64K) ..... \$12.50  
Deeper Dungeons (Req. Gauntlet) ..... \$12.50  
INFILTRATOR (64K) ..... \$12.50  
HARD HAT MACK ..... \$9.95  
TRAILBLAZER by Mindscape ..... \$12.50  
SPORTS SPECTACULAR 3 Games ..... \$9.95  
Dungeon of Deepair (3 Adventures) ..... \$9.95  
Flight Simulator Japan Scenery Disk ..... \$9.95  
Flight Simulator San Francisco ..... \$9.95  
CROSS COUNTRY ROAD RACE ..... \$9.95  
GHOSTBUSTERS Disk ..... \$9.95  
SEVEN CITIES OF GOLD ..... \$9.95  
RACING DESTRUCTION SET ..... \$9.95  
ONE ON ONE BASKETBALL ..... \$9.95  
LORDS OF CONQUEST ..... \$9.95  
F-15 STRIKE EAGLE Disk ..... \$8.95  
TOP GUNNER COLLECTION ..... \$8.95  
SILENT SERVICE Disk ..... \$8.95  
VISCALC Spreadsheet ..... \$19.95  
SHOWDOWN HOCKEY ..... \$14.95  
MINIATURE GOLF ..... \$10.00  
FORT APOCALYPSE Disk ..... \$7.50  
ELECTRA-GLIDE Disk ..... \$7.50  
MIND MAZES (Educational) ..... \$7.50  
VIDEO VEGAS Poker & Slots ..... \$9.95  
STRIP POKER ..... \$25.00  
ACTION BIKER ..... \$7.50  
KICK START ..... \$7.50  
MONEY TOOLS (Financial Utility) ..... \$9.95  
DIG DUG Disk ..... \$9.95  
PUZZLE PANIC Disk by Epyx ..... \$9.95  
CEST'E LA VE Disk ..... \$9.95  
OLIN EMERALD (Jr., Adv.) ..... \$9.95  
FINANCIAL COOKBOOK ..... \$9.95  
SONG WRITER ..... \$9.95  
PREPPE AND PREPPE II (2 GAMES) ..... \$9.95  
BUCKAROO BANZI ..... \$12.50  
WARGAMES (Based on the Movie) ..... \$12.50  
DAY AT THE RACES (Horse Race) ..... \$12.50  
CHESSMASTER 2000 ..... \$12.50  
PINBALL CONSTRUCTION SET ..... \$12.50  
BRIDGE 4.0 ..... \$9.95  
PEGAMMON ..... \$12.50  
MUSIC CONSTRUCTION SET ..... \$12.50  
FOOBITSKY Disk / Board Game ..... \$9.95  
TEMPLE OF APSHAI ..... \$9.95  
MOUSE QUEST Disk (64K) ..... \$12.50  
ELECTRONIC DRUMMER Disk ..... \$12.50  
NICKERBOCKER Disk ..... \$12.50  
12 SCOTT ADAMS ADVENTURES ..... \$12.50  
HACKER Adventure by Activision ..... \$9.95  
MIND SHADOW Adventure 64K ..... \$14.95  
BLAZING PADDLES (Graphics) ..... \$14.95  
SCHEMATIC DESIGNER ..... \$24.95  
THE SCANALYZER ..... \$29.95  
Advanced Protection Techniques ..... \$24.95  
BASIC TURBOCHARGER ..... \$23.95  
CHIMPUNK ..... \$34.95  
YOUR ATARI COMES ALIVE ..... \$23.95  
PAGE DESIGNER ..... \$15.00  
TYPESETTER ..... \$20.00

### SSI QUALITY SIMULATIONS

SSI U.S.A.F. .... \$19.95  
SSI COMBAT LEADER ..... \$19.95  
SSI PANZER GRENADEIER ..... \$19.95  
SSI FIFTY MISSION CRUSH ..... \$19.95  
SSI SONS OF LIBERTY ..... \$19.95  
SSI SIX GUN SHOOTOUT ..... \$19.95  
SSI KNIGHTS OF THE DESERT ..... \$19.95  
SSI Rebel Charge at Chickamauga ..... \$14.95  
SSI SHILOH - GRANTS TRAIL ..... \$14.95  
SSI EPIDEMIC ..... \$19.95  
SSI WIZARDS CROWN ..... \$14.95  
SSI GEMSTONE WARRIOR ..... \$14.95  
SSI COMPUTER QUARTERBACK ..... \$19.95  
SSI WARSHIP ..... \$19.95  
SSI NAM ..... \$19.95  
SSI WARGAME CONSTRUCTION SET ..... \$14.95  
SSI SHATTERED ALLIANCE ..... \$19.95  
SSI GETTESBURG ..... \$29.95  
SSI BATTLE OF ANTIETAM ..... \$29.95

### EDUCATIONAL DISKS

MASTER TYPE (Typing Tutor) ..... \$15.00  
UNKNOWN SPANISH ..... \$22.50  
UNKNOWN GERMAN ..... \$22.50  
UNKNOWN FRENCH ..... \$22.50  
A+ SPELLING ..... \$12.95  
A+ LEARN TO READ (3 DISKS) ..... \$12.95  
A+ READING COMPREHENSION ..... \$12.95  
A+ US GEOGRAPHY ..... \$19.95  
A+ FRENCH VOCABULARY ..... \$19.95  
SAT VERBAL ..... \$17.95  
COMPUTER AIDED MATH PROG. ..... \$10.00  
HODGE PODGE (AGE 1 - 6) ..... \$13.50  
MONKEY MATH / MONKEY NEWS ..... \$17.50  
TEDDY BEAR Rainy Day Games ..... \$12.50

### VIDEO TITLE SHOP - Create

Dynamic Graphics displays for shows, expositions, window displays or for use with your VCR. Features include text fade in and out, fine text scroll, crawl and more. Includes paint program and 2 graphics disks ..... \$14.95

CALL TOLL FREE

1-800-551-9995

OR CALL 510-352-3787

## AMERICAN TECHN-VISION

Mail Order: 15338 Inverness St., San Leandro, Ca. 94579

Sales Office: 1903 Fairway Drive, San Leandro, Ca. 94577

Terms: NO MINIMUM ORDER. We accept money orders, personal checks or C.O.D.s. VISA, MasterCard okay. Credit cards restricted to orders over \$14.50. No personal checks on C.O.D. - Shipping: \$4.75 shipping and handling on orders under \$150.00. Add \$4.00 for C.O.D. orders. In Canada or P.R. total \$8.00 for shipping and handling. Foreign shipping extra. Calif. residents include 8 1/4 % sales tax. All products new and guaranteed. All sales final. Send S.A.S.E for free catalog. Prices subject to change without notice. Atari is a reg. trademark of Atari Corp.

# TABLE OF CONTENTS

## FEATURES

- Us Guys Iz Yooze Guys** 6  
Managing Editor Ben Poehland describes recent events, their impact on AC's future, and the philosophical significance of this publication.
- Cheep Talk Type 'n' Talk** 19  
Louis Marcotte, AC Contributing Author, makes his debut with a machine language text-to-speech program for enthusiasts of Lee Brilliant's Cheep Talk speech synthesis module.
- A Look At Synchronesh** 29  
Reviewer Gary Matteson examines one of the rarer forms of high speed disk I/O for aficionados of Indus GT disk drives.
- Advanced C Programming On The 8-Bit, Part 2** 24  
Contributing author Mark Miller returns to wind up his excellent treatise on CC8 with some limitations of the program and some C programming examples.

## COLUMNS

- Exploring the Wild FONTier** 16  
Columnist Dave Richardson continues his exploration of Daisy Dot III with advice on double-column printing and print-spacing tricks.
- The Fitting Room** 10  
Columnist Mike Jewison describes "Wizzy", a delicious little 256K RAMBO clone that's dirt cheap and works like a charm.
- The Garret** 13  
Ed Hall, AC Staff Columnist, introduces us to some excellent text adventures from the British software developer, Level 9.
- Moonlight Workshop** 21  
Columnist Jeff McWilliams returns with the beginning of a multi-part series on RAM upgrades.
- Dollars and Cents in Education** 12  
AC's Educational Software Editor discusses educational software from Canadian distributor Bellcom and offers a brief review of "The Noisy Giant".

## DEPARTMENTS

Letters	4	Subscription form	32
Tips 'n' Tricks	9	News Roundup	30
Swap 'n' Shop	31	Dream Street	5
Correspondent's Corner	15		

## ATARI CLASSICS

Vol. 2 No. 2 April 1993

A Bimonthly Publication of:  
Unicorn Publications

Technical Consultant: Bob Puff  
Managing Editor: Ben Poehland  
179 Sproul Rd./Rt. 352  
Frazer, PA 19355-1958 USA

Magazine Staff:  
Circulation Editor: Ben Poehland  
Advertising Editor: Bob Scholar  
2535 Tamalpais Ave.  
El Cerrito, CA 94530

Hardware Editor & Utilities Software  
Editor: Charles Cole

1590 Foothills Drive  
Sierra Vista, AZ 85635

Educational Software Editor:

Bart Bresnik  
555 Ware St. Mansfield, MA 02048

Graphics & Entertainment Software  
Editor: Jeff Potter

814 Banbury Drive  
Port Orange, FL 32119

Telecommunications Editor:

Lawrence Estep  
524 Roseview Terrace  
New Albany, IN 47150-4457

Disk Editor: Steve Hoffee  
333 Peninsula Drive  
Lake Almanore, CA 96137

Disk Distribution Manager:

Alex Pignato  
Assistant Disk Distribution:

Jon Mordosky  
Staff Columnists: Ed Hall,  
Mike Jewison, Jeff McWilliams,  
Dave Richardson

Contributing Authors:  
Jeff Miller, Louis Marcotte

Staff Reviewers: Bob Hardy, Gary  
Matteson, Alan Sharkis,  
Bob Woolley

Correspondents:  
Itay Chamiel (Israel)  
Dean Garraghty (British Isles)  
Don LeBow (Hawaii)  
Marek Tomczyk (Germany)

**Manuscripts:** Please send manuscripts directly to the appropriate Editor. For a free AC Author's Guide send a written request to the Managing Editor. AC will pay \$25 for major Feature Articles upon Publication.

**Subscriptions:** See the Subscription Form elsewhere in this issue.

**Back Issues:** \$3.00 mag or disk, \$4.00 outside USA, payable to Ben Poehland in US Funds.

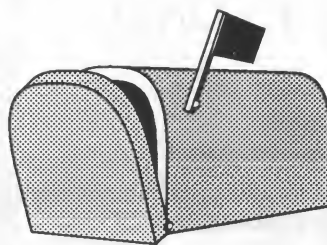
**Advertising:** Write to the Advertising Editor for a schedule of commercial ad rates & discounts.

Opinions expressed by authors are their own & do not necessarily represent the views of Unicorn Publications or the Staff of AC. Atari, the Fuji symbol, & names of Atari products are trademarks of Atari Corp. Other trademarks are property of the respective manufacturer.



## READERS' FORUM

### LETTERS TO AC...



Sir:

What is your FAX number? Postal mail takes a long time to and from New Zealand!

B.G.W. Dreaver Auckland, New Zealand

*AC Replies:* The "Editorial Offices" of AC consist of part of a small upstairs spare bedroom in my home. AC is a shoestring operation, run by a few dedicated part-timers. Sorry, it will be long time before we see a FAX machine around here. Postal mail will have to do. -BP

Dear Ben,

My AC software disk arrived a couple weeks ago. I was wondering what I would do with it when, yesterday, lo and behold my Feb. AC arrived in the mail. I was glad for the explanatory index!

The mag still looks good. Suggestion: don't be conned into going for glossy pages, fancy covers, etc. Keep it clean, simple, and inexpensive as possible. I prefer the present appearance to the fancy stuff.

How did that "ST or Amiga" bit get in there (Feb. issue, p. 13—side B, Multi-Mouse Driver)? Here I thought we were turning our backs on 16-bitters and not speaking with them! The next thing you know they will taking over as they always have in the past! Gloom, despair, and agony, oh me!

RE: the AtariWriter-80 bug ("Tips 'n' Tricks", p. 6, Feb. AC), I hardly consider this to be a "bug", since it can be a very useful feature, although it is an annoyance. An "S" or "W" would have been a better choice for that command. The problem, however, is not new with AT-80. It's a carryover from the 'Writer ROMcart and 'Plus versions. I've been exasperated countless times by the problem.

RE: "Advanced C Programming" on p.30 of the Feb. issue, for CC8 I saw no mention of real or floating-point variables. I hope this is not still another C that omits real arithmetic. Deep Blue C supposedly can use a Math Library disk. Mine, from ANTIC, would not work. Without real arithmetic, the DBC was worthless to me. If CC8 has FP variables, I surely would be glad to see it appear on the AC disk, even if it used the entire disk.

Here's a tip for ye olde FORTRAN programmers. In BASIC one can emulate the FORTRAN IF statement:

IF (variable) A,B,C

by using the following BASIC statement:

ON (SGN(variable))+2) A,B,C

My regret is that it took me so many years to figure that out! Here's to many more AC's!

Joseph Eichelberger  
Pinetta, Florida

*AC Replies:* Well, Joe, glad you got your disk OK, sorry we don't have much control over the quirkiness of the mails (but we're working on it!). The disk goes 1st class (which is fast), and the mag goes 3rd class bulk (slowest of the slow), and ne'er the twain shall meet. We have no plans to change the appearance of AC in the foreseeable future. We keep a pretty close eye on our bottom line around here (which isn't difficult since our bottom line is already pretty close to the bottom!). On the Feb. disk, the "ST or

*Amiga" refers to software drivers that were furnished for folks who might want to use an ST or Amiga mouse on their Classic Atari. As stated in the Dec. 1992 issue, AC's philosophy RE: the ST world is one of peaceful coexistence. The computer world is much bigger than us, and none of us can afford to ignore other platforms. However, you need not fear a takeover of AC by the "big" computers. AC was founded as an 8-bit-only magazine; it will live—or die—as an 8-bit magazine (see my article elsewhere in this issue). I'll refer you to Part 2 of Mark Miller's treatise on C programming (elsewhere in this issue) for your concerns about FP variables in C. As to whether we'll put CC8 on our disk, don't hold your breath. Our disk has proven to be a runaway success (over 250 subscribers), and so far you're the only person who's expressed interest. -BP*

Dear Classics,

I received your Dec. 1992 issue and was very impressed so I decided to subscribe. My check is enclosed.

I think I'm the only Atari 8-bit user left in Korea. News is scarce. One tip: you can use the Sega joystick with the Classic Ataris. Looks good, handles really well, matches the 800XL case, and only costs \$15.

Since you seem to know all the best programmers left in the 8-bit world let me give you my wish list. I'd like to see a patch to let me use PaperClip and SynCalc with my XEP-80 box. I hate AtariWriter and so far my XEP-80 just sits on the shelf gathering dust. Another thing I'd like to see is the porting of some of the old Commodore stuff. I'd like to see 4th and Inches, and also a football game for my machine that shows off its capabilities. Hardball was ported over; it doesn't seem like it would be that difficult.

Anyway, keep up the good work!

Dave Harris  
Seoul, Korea

*AC Replies:* OK, Dave, glad AC was able to provide a lifeline for your Classic Atari out there in the Far East. Thanks for the tip on Sega joysticks! As to patches for your XEP-80, I'm not sure, but I think we will have some XEP-80 drivers appearing on a future software disk, maybe you'll find them helpful. When it comes to porting games from other machines, that opens this whole hairy business of software legalities which hangs over the remnants of our community like a dark cloud. Considering that the Profit Motive has largely vanished as a major driving force in our market, I don't think it will happen. -BP

Dear Ben,

I was very pleased to receive my first subscription issue (#2) of AC this week. I trust response is growing and we can look forward to a full year's worth of AC.

I've recently assumed the role of Mailbag Editor for the U.K.'s "New Atari User" magazine published by Page 6 Publishing. Recent letters have arrived from various parts of the U.K. but also from Germany, Holland, and New Zealand. I look forward to more details of the inter-

national XL/XE scene in future issues of AC.

The mention of Dresselhaus' Dots Perfect upgrade to older Epson DM printers in the "Tips 'n' Tricks" column was especially interesting. I'll be enquiring of Epson U.K. if they can supply this add-on to tune up my RX80FT+.

I'm not a hardware hacker, but I do find columns such as Mike Jewison's "Fitting Room" of interest—who knows, there may be a hardware mod I can't resist trying!

Ed Hall's "Garret" column confirmed the fact that the Classics are not dead—there seem to be more disk mags than ever. I heartily recommend EXCEL and FUTURA to prospective buyers.

I was slightly disappointed to find Alan Sharkis' review of SIO2PC in the same issue as this product was described by Mike Jewison—it felt as though we were being cheated out of an article on something else. It was a competent article but perhaps should have appeared in a separate issue?

I'm very pleased with Dave Richardson's "Wild FONTier" column. This will encourage me to experiment more with the superb Daisy Dot III.

After reading Bob Hardy's interesting Keycode article, it seemed Bob apparently couldn't find a detailed list of keycode values anywhere. After checking through my collection of Atari bibles, I located the keycode values in Appendix A of COMPUTE!'s *Third Book of Atari* (1984). I'd advise any Atarian to get these (presumably now out-of-print) COMPUTE! books if you can find them!

Alex Pignato's article on User Clubs was a thoughtful piece—I hope it helps generate more enthusiasm in the 8-bit community. I'm almost tempted to start a "correspondence" disk—not so much for articles but rather an interchange of correspondence. I wonder how practical that would be?

I note the mention of Ace C in Mark Miller's article on C programming. A recent correspondent to the Page 6 Mailbag complained he couldn't get anything to work with the copy of Ace C he purchased from the Page 6 PD library. I'm asking for reader feedback on this as I find it hard to believe this is the case and I would imagine

Mark would have mentioned something of this in his article. If you have an address for Mark we'd like to enquire further.

Thanks for a great start to AC. I look forward to many more issues!

Allan J. Palmer

97 Pakenham Road

Basingstoke, Hants RG21 1YA ENGLAND

*AC Replies: Whew! I never quite envisioned AC would stand in the glare of a full-fledged literary review. I gather we have passed muster and that AC has found favor on the shores of Albion. AC continues to add new subscribers at the rate of about 45/month, and I'm hopeful we'll reach our quorum of 500 soon. Judging from the length of your letter and wide variety of topics you addressed, Allan, I don't think we have much to fear that two articles on SIO2PC diluted our coverage. In the belief that people write best on the subjects they feel most passionate about, I place very little restriction on topics covered by our authors, as there is already an inherent diversity in the community from which AC draws its manuscript material. I am in fact overjoyed on those occasions when two authors cross paths, as I revel in the prospect of seeing a product described from two different perspectives. We have in fact received several communications alerting us to the existence of the keycodes in reference material as you describe. It's rather discouraging that this information wasn't presented in any of the material furnished by Atari Corp., and Bob's program is no less useful for having been published and brought the matter to light. I would discourage your proposal for a communication disk (it would be way too slow) and recommend instead active participation in telecommunications. Commercial net services are available in the U.K., as well as the Internet (including the Info-Atari8 Digest and the USENET Newsgroup) and the British JANET gateway. The entire AC "movement" was literally born in a modem, and we feel telecommunications are essential to the survival of the worldwide 8-bit community. I've passed a copy of your letter to Mr. Miller with a suggestion to contact you RE: Ace C. - BP*



## Dream Street: Products We'd Like To See

(Not much change this time. Still some good opportunities for hackers, programmers, developers, and entrepreneurs.)

- New protocols for BobTerm: KERMIT, ZMODEM, and ANIS.
- A 24-pin Epson driver for Daisy Dot III with correct aspect ratio.
- A serial (SIO) buffer for slow printers like the 1027 and 1025.
- A PAL programmer.
- An adapter to convert composite video to TTL (MDA, CGA or VGA).
- A really good VT-100 emulator for BobTerm.
- An 8bit version of the ST program Aladdin for GENie users.
- A high quality electronic schematic drawing program with a 24-pin driver.
- A patch program to make AtariWriter-80 completely compatible with SpartaDOS.
- A utility for converting Print Shop fonts to Print Power format.
- A 256K upgrade for the 600XL (perhaps utilizing 44256 DRAMs).
- A drawing utility for Print Power.

Got an idea for a product? Send it in! If we think it looks good, we'll add it to this list!

# Us Guys Iz Yooze Guys: A Status Report On the AC Experiment

Ben Poehland, Managing Editor

## Go Ahead, Sniff It

Beginning with the February '93 *AC*, our Publisher switched to a different printer. That change was marked by two serendipitous events: first, the new printer didn't whittle down the page size as much, so we got a slightly larger magazine without any material being sliced off in the margins; and second, the new printer uses some kind of ink that smells really great. Yup, I'm one of those people that compulsively sniffs fresh print. My first impression of the printed copy of the February issue was, Gosh this thing not only looks great, it even *smells* good! I sure hope this April issue gets printed by that same printer using that same ink. Next thing you know I'll be eating the darn thing.

So go ahead, give it a sniff. Then give it a good read. Twice over, if you're the sort who needs extra time for digestion. Hopefully, there's a little something in here for everyone- whether you're a reader or a sniffer.

## What Is It?

Okay, we've established that this thing you're holding in your hands *looks* like a magazine and *smells* like a magazine. A magazine like *ANTIC*, or *Analog*, or *Compute!*, or *Current Notes*, or *AIM*, or *8:16*, or *Page6 New Atari User*- right?

*Wrong. AC is not like any of those magazines.* Some of the publications listed above are (were) commercial operations, founded by their respective proprietors for the purpose of making a profit. Commercial magazines are the property of their publishers: the magazine is the publisher's product which he offers for sale, and thereby earns a living for himself and his employees. Some of the magazines above have their origins in the usergroup community, having been founded as usergroup newsletters which grew to proportions approximating the size of commercial operations. At least one of them actually became a commercial operation, while others maintain a quasi-commercial status as non-profit operations with close ties to the usergroup communities from which they evolved.

So where does *AC* fit in? Nowhere, really. *AC* isn't a commercial operation like *ANTIC* or *Compute!* We aren't in business to make a profit, and none of our "employees" (known affectionately as "The Staff") is paid a salary. We are basically a non-profit organization (such profit as may accrue officially goes to our Publisher, but as a practical matter is actually returned to the "kitty" to seed future operations), so in that regard we do have something in common with the usergroup-originated publications like *Current Notes*. But unlike them, *AC* doesn't have any particular geographical point of origin, nor does it have any particular relationship to a usergroup.

The only constituency this magazine has is the worldwide 8-bit user community as a whole. *AC* is parochial to the extent that its language and style reflect the fact that it's produced and printed in the United States, but that's only because when you do something you have to be *some-where*. In theory, the idea of *AC* could have sprung up in any part of the world where people are still using Atari 8-

bit computers and didn't want to end the love affair we have with these old machines.

## A Bold New Experiment

I deliberately used the phrase "...the idea of *AC*..." in the preceding sentence to soften you up for the main point of this article. I've said this a thousand times to people already, and I'll say it here again:

*AC* isn't just a magazine. It is an Idea. It is a *bold new experiment*.

Memorize that. Burn it into your brain. Sniff the words off the page if you want. But never forget it.

Now I see some of you are confused, and with good reason. There has been quite a lot of confusion in the 8-bit community as to what *AC* is supposed to be. Obviously *AC* represents an idea, but what idea? What exactly is this Bold New Experiment?

In simplest terms, the Bold New Experiment consists of the Idea that a disenfranchised user community can preserve itself from oblivion by means of some vehicle that focuses the collective resources of the entire community. In this case, the "vehicle" selected was a print magazine. That magazine is what you now hold in your hand, and it's called *Atari Classics*. Quite literally, *AC* is the embodiment of the expressed will of the worldwide 8-bit community that it does not want to die.

Every time an issue of this magazine gets printed and distributed, the Atari 8-bit community is telling the world—and the dodo-brains in Sunnyvale—"We are still here. We haven't died yet. We are still a viable community of users who feel there is value in these machines. You can disown us, or laugh at us, but you can't kill us." To my knowledge, no other user community has ever done anything quite like this. As such, *AC* represents something new in the world: the idea that a user community can exist as an entity independent of the company whose product drew us together in the first place. We are breaking new ground here, and the whole world is watching.

## The Experiment Begins

Some of you may recall that in the latter half of 1991, Atari Corp. announced its intention to discontinue manufacturing and support of all 8-bit computer products. In the waning days of that year the future of our machines—and our community—became a raging debate on the networks, especially the Internet. Many felt that Atari was trying to kill us off, or coerce us into dumping the 8-bits in favor of the ST. Resentment was strong, but there was also consternation over what to do about it. Proposals of all sorts were hotly debated. It was the proposal made by Jeff McWilliams (now *AC*'s "Moonlight Workshop" columnist) for a mail-in campaign to start our own all-8bit periodical, that withstood the rigors of debate and emerged as the grass-roots movement that brought this magazine into existence.

Jeff's Mail Campaign was international in scope. Its goal was to find 500 people who would commit themselves in writing to subscribe to a magazine that didn't yet exist.



Jeff acquired five other people—dubbed the "Campaign Committee" (myself among them)—to assist in distributing about 1600 market survey forms to 8-bit users all over the world. The Campaign ended successfully in May of 1992, exceeding its goal of 500 commitments by a healthy margin. Jeff McWilliams became the Keeper of a Trust.

### The Experiment Takes Form

In retrospect, I think Jeff and his Committee were rather astonished at the results of the Campaign. Especially in the beginning, I don't think any of us really thought we'd meet our goal. There were about 1,000 people who said NO to Jeff's idea—which was mostly what we expected. The surprise was that about 600 people (615 to be precise) said YES, and gave their personal signatures on officially postmarked replies to back it up. During the Campaign everyone was focused on the daily logistics of getting out the survey forms and counting replies; no one really thought about what we would do if the hairbrained scheme actually *worked*. In May of 1992 Jeff suddenly found himself obligated to 600 rabid 8-biters who expected him to carry through the promises of the Campaign and start a magazine. As a struggling college student the rigors of the Campaign had stretched Jeff's personal resources to the limit, but the prospect of starting a magazine was totally beyond his ken. Recognizing the limits of his capacities, Jeff turned over the Campaign postcard responses to me and asked me to start the magazine.

It was in this way that I inherited The Trust and accepted the obligation for proceeding with The Experiment. Over the summer of 1992 I made arrangements with Unicorn Publications and devised a business plan for the magazine. Known as the "Atari Classics Publication Manifest", that plan is the foundation upon which the organization of this magazine is built. It's a long, tedious document. In it I drew upon every personal observation I had ever made about magazines. Mediocre magazines that survived. Good ones that failed. Forged in the cauldron of a user community on the brink of extinction, this magazine had to be bullet-proof. There were 600 people depending on me, and time was running out. I knew that if I failed, this would be our last hurrah.

Unicorn Publications accepted the Manifest in August of '92, and a Staff was hastily assembled. The free Premier Issue of *AC*, which had been promised to the Faithful 600 by the Campaign Committee, was prepared in September, launched in October, and mailed out in November with a December 1992 cover date. In the business planning for the magazine, we assumed we would get 500 subscribers (the minimum required for a full bimonthly production schedule on the basis of spreadsheet analysis) just from those 600-odd participants in the Campaign. For good measure, I decided to stack the odds in our favor by directing our Publisher to print up an excess quantity of copies (2,000 total) which we would also distribute for free. By the time the last batch of freebies was mailed out on December 28, 1992, we had given away over 1700 free magazines. That effort represented the extreme upper limit of the combined resources of the *AC* Staff and Unicorn Publications. It was our best shot, and I was confident our 500 subscribers would be forthcoming. It didn't happen.

### The Subscription Dilemma

In December and January I was mostly preoccupied with production on the February issue and our first Soft-

ware Disk. In my dual capacity as Circulation Editor I continued to process incoming subscriptions, but with all the chaos of those days I wasn't keeping a running tally. I sensed we weren't up to 500, but I thought we were close. In late January I prepared all the mailing labels for the February *AC* (those of you with 1025 printers will recognize the print style), and also did an exact count. The result was disturbing: we were mailing out only about 350 copies of the February issue. Alarmed, I requested Unicorn Publications to cut back on the print run of the February issue. After servicing our paid subscribers, storefront distributors, and gratis distributions, I was left with precious few copies to service new subscription orders and back issue sales.

Fortunately new subscriptions are continuing to trickle in at the rate of about 50/month, and as of this writing (the first week of March), our subscriber base is up to 400. That is not a bad number, but it restricts what we can do. Certainly it's enough to keep *AC* going the rest of this year. It will permit us to print at least five issues in 1993. Nevertheless, *AC* is 20% undersubscribed, and there's a price to pay. We may have to drop an issue (probably August). We'll be stuck at 32 pages (I'd like to see 40). And worst of all for American subscribers, we'll still have to use this crummy 3rd Class Mail, aka "Pack Mule Express" (ever wonder where the term "snail-mail" came from?). The good old days when you could plunk down a hundred smackers for a 2nd Class mailing permit are long gone; the post office will nail *AC* \$350 for that privilege these days, and the record keeping they require would warm the heart of any former NKVD officer. We're willing to put up with the Guvernment hassles, but not without the Magic 500.

### Probing the Riddle

Perplexed by the sluggish response in the face of an effort that seemed a sure-fire success, I did a little poking around—jawboning with folks, snooping on nets and such—to see if I could gain insight as to what was going on. I made a rough check of the original 600 Campaign respondents and discovered that barely 200 of them had honored their commitment. Apparently the magazine itself wasn't to blame; despite some grumbling that it wasn't long enough, most readers gave the December *AC* high marks. And our February issue was received with even more enthusiasm. Quality control isn't the problem; we got us a good product here.

A major factor seems to be a "wait-and-see" attitude. Dumb, dumb, dumb. Too late, the folks who are waiting to see whether *AC* will "make it" are going to realize this magazine is for real. When they finally get around to honoring their pledges the back issues will all be gone, and they will miss out. We have to pay our bills in hard cash up front and can't afford to incur debt for costly print overruns in the hope we'll be able to sell the extra copies later to the slothful sluggards out there. I sent out an announcement to the networks explaining the consequences of our slow growth and the short supply of the February issue. Predictably, subscription orders increased by 40% in the following weeks. The small stack of spare copies of the February issue continues to contract each week as new orders go out the door, and now supplies of our first Software Disk are running out. I'm awaiting a final shipment from *AC*'s Disk Men, Steve and Alex, as they scrape the bottom of the barrel. When they're gone, they're gone.

Some complained about the slow delivery of the December issue. There wasn't much I could do about that

except offer my sympathies, and there's no question the slow deliveries delayed our subscription drive. The worst case of "snailmail syndrome" came from Iceland, where the November 3 mailout didn't arrive until February 5 (yeeesh!). Still, mailing delays alone don't account for the gap between expectation and performance.

Of all the reasons I encountered for why people were dallying, the one that concerned me the most was antagonism toward Unicorn Publications. Unicorn Publications faltered on deliveries of *Atari Interface Magazine* after October, which understandably annoyed a lot of people. Observing that *AC* was also published by Unicorn Publications, there were a number of cases where people took out their frustrations with *AIM* by refusing to honor their Campaign commitment to *AC*. The inference was drawn—quite incorrectly—that because *AIM* was in trouble, *AC* must follow.

With *AC* more or less up and running, people were now lumping it in with *AIM* and all the other magazines they had known, and treating it so. Former participants in the Mail Campaign had forgotten that *AC* was founded at their bequest and was being run by the 8-bit community itself, not by Unicorn Publications. And there were many newcomers who, totally unaware of how *AC* got started, had a cavalier attitude and were making all sorts of demands upon us as though they were dealing with some monolithic corporation. I realized we had a problem of perception, which motivated me to write this article.

## Who Owns AC?

By mentally linking *AC* with *AIM*—and thereby to all the other Atari magazines they might have known past or present—people were implicitly assuming that *AC*, like those other magazines, is the property of its Publisher. Unicorn Publications does indeed publish *AC*, but they don't really own it; at least not in the same sense that they own *AIM*. True, Unicorn Publications provides professional services to *AC* that are absolutely essential: DTP services, arrangements for printing and mailing, paying the bills, assuming obligations for legal and tax liabilities, and processing payments made with credit cards. But *AC* is basically a contract job for Unicorn Publications: we're paying them to do that stuff. Theoretically, we could have asked anybody to produce *AC* for us. I selected Unicorn Publications because I felt they had the most to offer, and from personal contact with Bill and Pattie Rayl I sensed a willingness on their part to support the 8-bit community.

In late February 1993 Unicorn Publications circulated a letter to all their *AIM* subscribers explaining their difficulties with that publication. Included with that letter was some commentary on the misperceptions people have about the relationship between *AIM* and *AC*, reassuring readers that the problems of *AIM* are completely unrelated to *AC*. I also noted with some amusement that Unicorn Publications referred to *AC* as "...Ben Poehland's new...magazine." The *faux pas* is certainly understandable. I'm the fellow Unicorn Publications deals with for 90% of *AC*'s business. And since I provided the loan that enabled Unicorn Publications to produce our first issue, it might be argued that I do indeed own *AC*. Even as I write these words, I still find the concept amusing.

The loan—which is interest-free—was nothing more than part of the machinery I created to fulfill the obligation I assumed when I accepted the legacy handed me by Jeff McWilliams. Sure, there was some personal risk to me in making that loan, but that went with the territory. Good grief, you have to start somewhere, and printers and postmen don't accept IOU's. A bit of "pump priming" was

required, and if *AC* lasts long enough I'll eventually break even. There is no personal gain to me from my financial arrangements with Unicorn Publications. That is as it should be: as I said at the beginning of this article, *AC* is not a commercial operation, and profit isn't our motive. I certainly do not feel authorized to claim ownership of this magazine, nor do I wish to.

I'm nothing more than the Keeper of the Trust that was bequeathed to me by Jeff McWilliams on behalf of those members of the worldwide 8-bit community who participated in the Mail Campaign. The sobriquet "Keeper of the Trust" has now been formally replaced by the title of Managing Editor. In that capacity I'm serving out my obligation to attend and supervise the Bold New Experiment represented by this magazine. In theory any member of the 8-bit community could step in and take my place. Indeed, if tomorrow someone appears who is better qualified than I, I would hand over this zany enterprise to him without hesitation. And the new Managing Editor would, in his turn, become the Keeper of the Trust.

I hold the opinion that the true owners of *AC* are those 200 people of the original 600 who followed through on their commitments to subscribe to *AC*. The remaining 400 are silent partners who haven't yet shown up to collect their piece of the action. The other 200 folks who signed up after the Campaign ended are owners too, but most of them don't know it. They think they merely subscribed to a magazine, and they are just now discovering that in reality they are fellow Participants in the Bold New Experiment. Thus it will be with everyone who subscribes to *AC* in the future: you aren't just a subscriber, you're a Participant.

## Us Guys Iz Yooze Guys

So who are these people on the *AC* Staff, and who do they work for? All of us are 8-bit users just like you. We all subscribe to *AC* just like you do, and we suffer along with you when it gets delivered late. We do receive some minor remuneration for out-of-pocket expenses (postage, phone bills, etc.), but none of us accepts a salary or compensation for our time and effort. We come from all walks of life and are scattered among several continents around the globe. This, incidentally, is why *AC* lacks a distinct geographical focus and bears no particular allegiance to any user group organization. Some of us belong to user groups; some don't.

We are all ages, from college students to fellows well past retirement. We represent a wide range of occupations: astronomer, chemist, military officer, lawyer, aspiring writer, musician, electrician, computer professionals, and even a guy who works in a gunpowder factory making artillery shells for the Army. Like you, we on the Staff spend our daylight hours earning a living in occupations unrelated to *AC*. It takes a certain amount of intestinal fortitude, or religious zeal or whatever, for us to create and assemble the material for this magazine in our spare time. Relative to the world of publishing, we're all a bunch of rank amateurs. We're still feeling our way, trying to avoid catastrophic errors and making dozens of little mid-course corrections to tweak our efficiency. Our logistics are a nightmare. Our status with the Infernal Revenooers is anybody's guess. And as with any collection of human mortals, we have our share of internal squabbles, disappointments, hurt feelings, and communication failures. But then somehow another issue of *AC* gets published, and the immense satisfaction we feel makes it all worthwhile. For better or for worse, warts and all, the *AC* Staff are the Caretakers of The Experiment.



We are motivated by the same impulses that drove you to chance a subscription to this novel enterprise. The authority for us to do the stuff we do is derived directly from *you*, and it is *you* that we work for. The material we publish is partly supplied by members of the community at large, and partly by Staff members. Since the Staff of AC is drawn directly from the global user community, it's senseless to distinguish between ourselves and the com-

munity we serve. *We are you.* No corporate entities here. No glass-and-steel office towers. No starched white shirts. Just a bunch of visionary guys with an incurable addiction to Atari 8-bit computers who want to see The Experiment succeed. Including one guy who sniffs fresh print.



## Tips 'n' Tricks

**Attention Beginners:** many new users have begun their Atari 8-bit computing interests with systems obtained second-hand from garage sales, swap meets, or family hand-me-downs. XL/XE systems obtained in this manner are nearly always bereft of manuals, leaving the novice user in a state of bewilderment upon arrival home with the newfound treasure. The situation is especially frustrating for the XL owner even if his second-hand system came with a manual: Atari's manuals for the XL machines were all worthless fluff, a pathetic joke. Fortunately, Atari's manuals for the XE machines were excellent, and AC recommends that anyone seeking basic user information on the XL/XE series computers should obtain the manual for the 130XE. That manual contains detailed descriptions of keyboard functions, pinouts of all the hardware ports, connection diagrams, and a splendid introductory tutorial on BASIC programming complete with a good-sized library of type-in programs. The information in the 130XE manual is 98% applicable to XL owners, the differences being in different physical locations of things like the function keys and various hardware ports, plus the differences in RAM capacity between the XL/XE. Try shopping around for this manual at Best Electronics (408-243-6950), B&C ComputerVisions (408-986-9960), Toad Computers (800-448-8623), or any of the advertisers in this magazine.

**MONITOR CABLES:** we've received several inquiries from people seeking monitor cables for their 8-bit. These cables are readily available from most of the advertisers in this magazine. From Dave Paterson, an 8-bit user on GENie, comes the tip that you *might* be able to get an Atari-compatible monitor cable from Radio Shack, catalog #42-2394. According to AC's Hardware Editor this item was deleted from the 'Shack catalog about five years ago, but you might luck out and find one still on the shelf if your local 'Shack store does a slow business.

**Atariwriter-80 Weirdness** (will it never end?): when running AT-80 under SpartaDOS 3.2d, if you're using an AUTOBAT file (for preventing, say, reformatting a RAM-disk when RESET is pressed), execution of this batch file upon pressing RESET will cause AT-80 to lock up upon loading. Until someone in the 8-bit community gets around to writing a SpartaDOS utility for AtariWriter-80 similar to the LVAUG Patch for AtariWriter+, the only cure is to remove (or rename) the AUTOBAT file from your AT-80 disk.

Another AT-80 oddity is seen in EDIT mode when the cursor is in the extreme upper left-hand corner. If you hit BACKSPACE, the poor little cursor goes blinky-crazy trying to go back one space even though there's no place for it to go (if there's text on the screen, it will commence re-writing the screen over and over). The cure is to just hit any character key. This will get the cursor moving in the forward direction it seems to prefer and relieve the hiccups.

**XF551 Socket Snafu:** recently one of the guys here at AC rediscovered- the hard way- an old problem with the SIO sockets on the XF551 disk drive. The controller board in the 551 is the crummiest piece of junk Atari ever made, cheap phenolic substrate with foil traces so thin you could probably see through them were it not for the green foil mask and sticky gobs of solder flux residue. To make matters worse, early production runs of the 551 used cheap rivets to fasten down the SIO sockets (Atari installed heftier sockets in later production units, fastened with screws). The rivets were undersize and incompetently installed, so the brunt of force applied when the user connects/disconnects cables is borne by the socket pins themselves. The pins are secured by solder connections to the thin foil traces, which soon work loose and break after a bit of wiggling by the user. The broken foil traces disrupt communication between the computer and the drive, giving rise to a host of devilish intermittent problems until finally the computer just gives up and refuses to acknowledge the presence of the drive.

The cure requires some skillful surgery (if you're the impatient/butterfingere type, find a friend with the necessary skills and equipment). Open the drive case and remove the controller card (you'll have to remove the drive mechanism first). Drill out the rivets from both sockets. This is best done working from the foil side of the board using a Moto-Tool equipped with an emery cutting wheel. Holding the card foil side up, gently wiggle the sockets while observing the SIO pin solder pads: a very strong light and magnifying glass will be indispensable here. Sometimes the pads themselves are lifted completely off the board, and foil breaks typically occur where the trace meets the pad. Make notes of which traces appear loose or broken, then fasten down the sockets with #4 hardware (screws, lockwashers, and nuts). Screws should be installed from the foil side of the board, with the washers and nuts visible on the component side. Then carefully re-solder all the broken pads and foil traces, taking care not to create solder bridges between adjacent pads. You might need to employ a razor edge to scrape the green mask off the trace to expose fresh copper for soldering; this requires considerable skill due to the delicacy of the foil runs. A short length of insulated wire can be used if the foil trace is mangled.

If the drive still misbehaves following this treatment, re-check your work. Sometimes the foil breaks are fiendishly difficult to see, especially with the unaided eye.

*We want bugs! If you have a hot tip on hardware/software usage or an undocumented bug in a program, we want to hear about it! Send your info to AC at our Editorial Office, 179 Sproul Road/Rt. 352, Frazer PA 19355 USA, ATTN: TnT.*





# The Fitting Room

## The RAM Thing

We've got a lot of stuff to cover this time, so let's get right to it. If you remember, I was looking for a memory upgrade for my 800XL. Not knowing a lot about the available upgrades for the XL I started browsing the literature for whatever RAM upgrade gems I could dig up. (After my experiences from last time, I decided to stay well away from any online service!)

### Thanks For the Memory!

Probably the best-known of the XL memory upgrades is the RAMBO XL, from ICD. Unfortunately, ICD dumped support for the 8-bit Atari computers beginning 1992- to the chagrin of many of us- with the result that RAMBO is difficult, if not impossible, to find anywhere. Although it was not completely 100% compatible with the extra memory in a stock 130XE, it did support BASIC XE's extended mode as well as RAMdisk handlers with Atari DOS 2.5, SpartaDOS, and MyDOS (and probably others, as well). It was also relatively inexpensive; \$39.95 not including the necessary RAM chips. At the risk of sounding nostalgic, I remember when I purchased a 48k memory card for my Atari 400 way back in 1982. It cost me almost \$400.

Computer Software Services, a company of which I am particularly fond, offers their Power Plus 256K memory upgrade for the 800XL for \$99.95. CSS claims this upgrade is the most XE-compatible available for the XL. Compatibility is an important issue for me, but it's not worth \$100 (not in my opinion, anyway). The advantage to the Power Plus is that if you send in your computer CSS will install the upgrade at no additional cost. This can be a boon if you happen to be all thumbs with a soldering iron.

Newell Industries offers a variety of upgrades, including a 1-meg behemoth. *[Editor's Note: I just installed one of these in an 800XL a few weeks ago; it's awesome! -BP]* One problem with this gigantic upgrade is that in order to use it you end up rendering the internal BASIC inaccessible from software. You can still gain access to it by wiring the BASIC enable line to an external switch and then boot with BASIC on or off, depending on the switch setting. I like the idea of having internal BASIC accessible from software (call it personal bias), so even though the idea of a monster RAMdisk is extremely appealing, I decided against it. Newell does offer a smaller 256k upgrade which is available from American Techna-Vision for \$27.50. By the time you add in the cost of the 256k RAM chips you're looking at a total cost of around \$47.50. I thought this sounded like a pretty good deal, but I kept poking around to see what else I could find.

I happened to be looking through my Best Electronics catalog shortly thereafter when I noticed something called the Wizztronics 256k memory upgrade. The bare board was just \$11 and Best was offering the RAM chips for \$14. There were a couple of other components which needed to be mounted on the board, but they came to just a couple of

### Mike Jewison, AC Staff Columnist

dollars. So for roughly \$25 I could upgrade my 800XL to 256k. I started thinking that there must be something wrong with this. Why was it so cheap? I called up Brad Koda, owner of Best, and asked whether or not he knew if MyDOS and/or SpartaDOS would properly recognize the Wizztronics; unfortunately, he didn't. So I threw caution to the wind and ordered all the components there on the spot, thinking that if it didn't work I'd only be out \$11, because the RAM chips could probably be used in some other (more compatible) upgrade. About a week and a half later, the package arrived.

### Resistance is Futile

The documentation for the Wizztronics, affectionately known as "The Wizzy", is scanty at Best (pun intended). A single sheet of paper accompanied the board indicating the connections between the Wizzy and the motherboard. The first thing you do is to pry the 64k RAM chips and replace them with the newly purchased 256k chips. The Wizzy plugs into a socket near the RAM chips via a double-sided header. There are either three or five components (TTL gate chips) which are inserted on the Wizzy board depending on the exact version of your Antic chip; I used sockets rather than soldering the chips directly to the board. You'll also need some scrap wire to make five connections between solder pads on the Wizzy and five pins on the 6520 PIA as well as one additional connection between the Wizzy and the motherboard. Since I hate soldering anything directly to a pin, I took a pair of tin snips and cannibalized a chip socket to give me five inline pins. I then soldered the five wires from the Wizzy to the hacked socket pins and plugged the PIA into them. This also makes for a real quick deinstallation should I ever decide to remove the Wizzy.

Everything was installed, save for one component, and it was here that I ran into a problem. The printed instructions call for a 33-ohm resistor to be soldered onto the Wizzy board. Hand-scrawled on the instruction sheet was a notation that resistor R32 on the motherboard is a 33-ohm resistor and this resistor must be removed in order for the Wizzy to function properly. (R32 sits immediately behind the bank of RAM chips.) The logical thing to do, then, was to take the resistor which had just been removed from the motherboard and insert in onto the Wizzy. The only problem is that when I looked at the resistor I'd pulled, it was 3100 ohms— not 33! So now I was in a bit of a dilemma. The printed instructions said to use a 33-ohm resistor, but the hand-written note told me to use a 3100-ohm resistor pulled from the motherboard. With great trepidation I decided to follow the hand-written instructions and use the one I'd removed from the motherboard. Big mistake.

When I finished the installation, I powered the computer up (without the disk drive) and was greeted with the UltraSpeed+ OS's information screen. (The

UltraSpeed+ is a replacement OS module for the XL/XE computers, available from Computer Software Services.) The US+ includes its own set of test utilities and I figured I should probably use those to test the machine, particularly memory. First off, I knew I had a problem - the US+ thought I was running a 192k machine, not 256k. In a valiant attempt to rationalize this, I thought that perhaps this number referred only to memory above the stock 64k. As I started the tests to check RAM bank by bank, it was clear something was wrong. The OS would get into the first bank of 16k and spit out an error message. Every time. I consulted with a couple of people and decided to remove the 3100-ohm resistor from the Wizzy and replace it with a 33-ohm unit. I made the change and powered it back up again.

This time, the US+ told me I had a 256k machine. And the RAM tests turned up no errors at all; all 12 banks of extended memory passed the various checks with flying colors. Hot dang - it worked!

Upon further consultation with assorted gurus and Alchemists, I decided to perform a little experiment; I re-installed resistor R32 to its original location on the XL motherboard. This resistor, I am informed, is a pullup resistor for the REFRESH line on the RAM chips and one of its functions is to reduce ringing and help stabilize operation. One or two people told me that it should not be necessary to remove it from the motherboard to have the Wizzy perform properly. You know what? They were right! With R32 back in place the upgrade performed just as it had with no resistor.

So why didn't the upgrade work with the 3.1k ohm resistor on the Wizzy board (rather than the proper 33 ohm)? Our best guess is that the value of this resistor is so high that it's probably dropping the signal voltage when the REF line is pulsed. Needless to say, the last thing you need with RAM chips is to have the signal voltage drop. Depending on the amount of voltage reduction, the RAMs may or may not get refreshed; the result is something resembling the HAL 9000 at the end of "2001: A Space Odyssey" (i.e. real screwed up).

## Acid Test

Now that I had a functioning memory upgrade installed it was time to test it out. I had three criteria the upgrade had to pass: it had to function properly as a RAMdisk for (a) MyDOS, (b) SpartaDOS (both 3.2 and the X-cartridge), and (c) the UltraSpeed+ OS RAMdisk handlers.

To make a long story short, I was able to successfully create usable RAMdisks with all three of the DOSes mentioned. The RAMdisk created with SpartaDOS-X is 16k smaller than the others (176k vs. 192k) because SDX reserves a 16k bank of memory for its own nefarious purposes.

The final test to run was to check the RAMdisk handlers with the US+. I had no reason to doubt that it would work, particularly since the US+ was able to earlier recognize the fact that I had a 256k computer. As you will undoubtedly remember, I was planning on using the US+ to install the RAMdisk as drive 1 and run my beloved Infocom games from there, speeding up the games and saving wear and tear on my now ancient Percom disk drive. In order to accomplish this, however, there were a number of preliminary steps I had to perform:

- 1.) Get into the US+ menu and install the RAMdisk as drive 1 and the Percom as drive 2.

- 2.) Use the US+ sector copier to copy the game data (side 2 of the disk) from floppy to the RAMdisk.

- 3.) Temporarily boot from drive 2. This allows the computer to boot from a drive other than drive 1.

- 4.) Once prompted to flip the disk, hit CTRL-SHIFT-6. This restores the drive configuration to the US+ settings (in other words, the floppy becomes drive 2 again). When you press RETURN at this point, you are into the game with all disk access now occurring from drive 1 (the RAMdisk).

I played Enchanter (one of my favorites) for about an hour and was able to get much farther into the game than I normally could in a one hour session. Needless to say, I was ecstatic that the memory upgrade had met all my criteria. I then decided to try a little experiment. I made a list of 20 commands (usually the first 20 things I do when I start Enchanter from scratch) and timed how long they took. When I ran the game from the floppy, it took about 115 seconds; from the RAMdisk that number was reduced to 90 seconds. If you consider that I likely spent about 50 to 60 of those seconds typing in the commands, the true increase in speed resulting from the use of the RAMdisk is on the order of 2 to 3 times. Applications such as database programs, which can be very disk intensive, will likely see even greater gains. With this kind of increase in productivity, I'm looking forward to trying out a number of other applications and games with my US+/RAMdisk combo.

## Go Ahead, Make My Day...

So what are we left with here? Now that the dust has settled, it would seem the Wizzy is a direct replacement for the recently departed RAMBO. And not only is it a replacement, but a much less expensive one at that. *[The Wizztronics upgrade is actually a bare-bones RAMBO "clone" which, like RAMBO, traces its origins to the article published in BYTE magazine in 1985 by Claus Bucholz. - BPJ]* Keep in mind, though, that if you decide to install the Wizzy yourself there are a couple of errors on the installation sheet:

- 1.) The resistor to be installed on the Wizzy daughter-board should not be taken from the motherboard. Regardless of what's written on the instructions, R32 is NOT a 33-ohm resistor. Find a proper 33-ohm resistor from another source to use with the Wizzy (Radio Shack part #271-007, 29 cents/pkg. of two).

- 2.) Do *not* remove resistor R32 from the motherboard; it's not necessary. Any time you can perform an upgrade without having to remove something, you have simplified it by an order of magnitude!

That said, I *heartily* recommend the Wizztronics 256K. I've been using mine for the better part of a year and, since the initial installation, haven't run into any problems with it.

Rats. Now that I think of it, it wasn't Rambo who said "Go ahead, make my day", it was Dirty Harry. Sigh. I need a vacation!

Next time: Christmas comes to the Fitting Room. In June???

## Items mentioned:

Wizztronics 256k Upgrade Best Electronics 2021 The Alameda, Suite 290 San Jose, CA 95126-1127 U.S.A. (408) 243-6950 Bare board: \$11.00 RAM chips: Call



UltraSpeed+ OS Computer Software Services P.O. Box 17660 Rochester, NY 14617 U.S.A. (716) 429-5639 \$69.95

*Note on 256K RAM chips: the type of chip used in all 256K upgrades is the 41256 dynamic random access memory (DRAM), a 16-pin DIP that's pin-for-pin compatible with the original 4164 (64K DRAM) chips in your stock machine. If you do any kind of RAM upgrade, don't trash your original 64K chips; it's a real good idea to hold onto them. Prices on 256K DRAMs are now down to about \$1.25 each—or less—depending upon the "speed" of the chip. Speed is measured in nanoseconds (nS). The faster*

*the chip, the higher the price. Typical speeds are 60, 80, 100, 120, 150, and 200nS. Fortunately, Atari 8-bit upgrades work best with the "slower" (cheaper) chips in the 100-150nS range; the 200nS chip- if you can find one—will probably work but might lie on the cusp of reliability. The faster chips—below 100nS—will also work but gain you nothing except to waste your money; prices on those can range beyond \$2.00 each. Check out an issue of Computer Shopper magazine (\$2.95 off the rack at most US outlets; also available in Canada and the U.K.) if you really want the low-down on latest prices. - BP*



## Dollars and Sense in Education Inexpensive Educational Software

**Barton M. Bresnik, Educational Software Editor**

### The Bellcom Connection

If you're looking for inexpensive programs for education, Don Bell offers one of the most comprehensive Public Domain libraries available for the 8-bit Atari. His catalog, available in print with a supplementary disk, lists hundreds of titles. In reply to my request for information on educational programs, he wrote: "Enclosed is a copy of the BELLCOM 1992 Catalogue. The Educational section is found on pages 29-31 and includes...[such programs as Ancient] Civilization, Micro Tales, The Living Cell, and the Earth Sciences series. ...There are also other programs throughout the catalogue which could be used for educational purposes... [such as] Mandelbrot Sets, P.D. Paint, Paper Planes, Music Major, Learning To Program In Atari BASIC, and Assembly Language Tutorial."

Prices vary from \$2.95 to \$3.50 (U.S. dollars), depending on quantity ordered. Mr. Bell states he processes and ships within five days of receiving an order, though postal delivery may take three to five weeks. My own experience with orders from BELLCOM, placed recently and also a few years ago, has been reasonably prompt delivery of all items, with the inclusion of a "bonus" theme disk and an information disk. Though not all disks proved equally useful, they were all well described in the BELLCOM catalog. A number of the programs are now in use in my middle school classroom. Despite the Classic Atari's orphan status, BELLCOM continues to be an excellent source of educational software! For a catalog, write to:

BELLCOM Public Domain Software  
P. O. Box 1043  
Peterborough, Ontario  
Canada K9J7A5

### The Noisy Giant

The following brief review of Bellcom's "Microtale 1" disk was prepared by Ms. Anna Mahan, Instructional Learning Assistant, Coelho Middle School, Attleboro MA 02703 USA.

"The Noisy Giant" is available from BELLCOM as "Microtale 1", disk #154, which is part of a highly motivational public domain reading package. It provides high interest reading for the average fifth or sixth grade reading level. The illustrations- with bright colors, cartoon-like quality, animation, and sound effects- make this pro-

gram attractive for children in the elementary or middle-school age groups. The animals, people and flowers are an added attraction to the eye, helping to motivate students to continue reading and also helping to keep their attention focused on the story.

The plot is interesting enough to keep the reader's attention. An element of mystery is introduced when the giant finds the flowers destroyed and the animals all gone: children will read on to find out what happened. The vocabulary level is well-chosen, containing sufficient variety to maintain interest without imposing undue difficulty. Students exposed to this program will likely stay with it long enough for successful completion because the visual and auditory clues give powerful assistance to the child's comprehension of the text.

The introduction to the program documentation contains the following statement: "This software is FREE, compliments of the author - Charlie Parker. Please feel free to make copies for your friends and neighbors. If you like this software and would like to make a DONATION to the author, please write to: Charlie Parker, 14824 E. Colgate Dr., Aurora, CO 80014." Given the obvious effort that went into creation of "The Noisy Giant", it would be nice to know if Mr. Parker has written additional software of this type.

[Editor's Note: the address above has not been verified.—BMB]

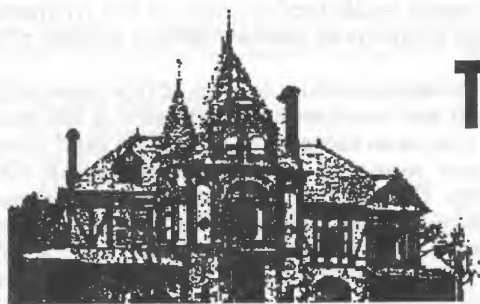


### The 2600 Connection

The newsletter for Atari 2600 Videogame owners, players, and collectors. Receive a sample issue of 100% coverage of the machine that started the home videogame craze! Each bimonthly issue contains news, hints, tips, reviews, solutions, and more.

Send \$1 for your sample issue to:

**Tim Duarte, Editor**  
**The 2600 Connection**  
**P.O. Box 3993**  
**Westport, MA 02790-0298**



# The Garret

## LEVEL 9

Ed Hall, AC Staff Columnist

### Text Adventures: A Fading Genre?

If you're interested in writing, you might have toyed with the idea of doing a text adventure. After all, it's composed entirely of words, and has occasionally attracted the attention of established writers like Thomas Disch and Douglas Adams. Synapse and Broderbund brought out a line of text adventures dubbed "electronic novels," while the folks at Infocom called their games "interactive fiction" and billed themselves as "the master story-tellers."

Unfortunately, text adventures have less to do with good prose than good puzzles, and today the genre seems all but defunct. What happened? Are text adventures dead forever, or can they be resurrected? In the months to come we'll examine these questions, and discuss tools and references for those interested in learning how to write their own. In the meantime, I'd like to profile a British company that produced some of the best text adventures on the market, yet remained poorly known on this side of the Atlantic. The company is Level 9 Computing, and it was begun in the early 1980s by three brothers: Mike, Nick, and Pete Austin.

### A Quick Overview

One of the most interesting aspects of Level 9's development was its support of tape drives. As far as I know, every adventure they produced was available on cassette, even its last one, which came out in 1989. This had a marked influence on program design, resulting in games which loaded entirely into memory and needed no further media access during play. It also made possible three other commands much appreciated by players: RAM SAVE, RAM RESTORE, and UNDO.

Later on, to get around the size limitations imposed by the use of cassettes, Level 9 began releasing games in the form of trilogies. These were either anthologies of older games, or newer ones broken up into three parts. For tape users, this meant three cassettes, while disk users got a single disk in 1050 "dual" density.

At the heart of any text adventure is a parser, and Level 9's is top notch, able to handle input such as:

EXAMINE ALL BUT THE WATCH, SPANNER AND TORCH AND GO EAST. DROP EVERYTHING BUT THE FLASK. OPEN IT AND GIVE IT TO THE ROBOT.

Early in its history, the company experimented with graphics, but finally resolved upon a course which reserved pictures for the newer, more powerful machines such as the Amiga and Atari ST. Level 9 ported its games to a wide variety of machines, provided comprehensive clue sheets, and in many games unleashed a superb sense of humor. Though packaging was not as elaborate as Infocom's (usually consisting of a novella which set the game's background), Level 9's prices were correspondingly lower.

Tracking the lineage of Level 9 games is something of a quest in itself. Initially the company released games under its own label, then subsequently enhanced and re-released many of them for Rainbird, which distributed them in both the U.K. and America. Eventually Level 9 grew dissatisfied with this relationship and switched to other distributors, Mandarin in the U.K. and Datasoft in America. (Datasoft never released any Atari 8-bit versions, even though they were available.) Level 9 produced its last text adventure in 1989. Declining sales finally convinced the Austin brothers to abandon the genre.

### Jewels of Darkness

Level 9's first text game was Colossal Adventure, a version of the "Original Adventure" written by Crowther and Woods on a mainframe. Level 9's version has 70 additional locations with a Middle Earth theme: an appropriate choice, since Tolkien's work had influenced the original game. Two more games, Adventure Quest and Dungeon Adventure, continue this theme. Later on, Level 9 added graphics, made a number of improvements, and released all three games in a single package. The name they wanted to use was Middle Earth Trilogy but their distributor, Rainbird, decided upon Jewels of Darkness.

Reception in North America was definitely mixed. ANTIC thought the games were top notch, *Computing Gaming World* said they weren't "completely bad," and *QuestBusters* excoriated Rainbird for not acknowledging Colossal Adventure's debt to Crowther and Woods.

### Silicon Dreams

This is another trilogy composed of three games originally issued separately. Kim Kimberly stars in Snowball and Return to Eden, while his/her descendant is the hero in Worm in Paradise. It all begins when Kimberly awakens aboard the spaceship, Snowball, which is headed on a collision course for the planet Eden. The ship's name is a double pun, for it refers both to the ship's cargo of cryogenically preserved humans, as well as to its chances for survival.

Graphics were originally incorporated into Worm in Paradise, but not (I believe) Snowball and Return to Eden. However, the latter two were retrofitted with illustrations when released by Rainbird as Silicon Dreams.

### Knight Orc

This game represents a turning point in Level 9's history. To begin with, it was the company's first attempt to make humor and fantasy key elements in a game—a successful formula which would be used again in the future. Secondly, Knight Orc seems to be Level 9's first tripartite game designed as a unit; none of its parts (Loosed Orc, A

Kind of Magic, and Hordes of the Mountain King) had been previously released. Thirdly, the game represents a departure in Level 9's handling of graphics. In previous games, the pictures were simple line drawings employing only a few colors—certainly a disappointment for people with 16-bit machines. With Knight Orc, Level 9 addressed this problem by furnishing illustrations which took full advantage of the more advanced machines. At the same time they eliminated graphics from the Atari 8-bit version (and possibly did the same for the Apple II and the Commodore 64 versions). Finally (on a negative note), though Level 9 produced an 8-bit version which Rainbird released in England, Knight Orc (as far as I'm aware) didn't make it to these shores. Nor, to my knowledge, did any subsequent Level 9 game for Atari 8-bits.

Knight Orc begins with a wonderful role-reversal, casting the player as one of the traditional bad guys of fantasy, an orc. In fact, so despicable are you that even when you die, you are quickly turfed out of heaven (and returned to the game, of course). As you make your way through the various scenes, you hear the plaintive cries of other hapless adventurers, some pleading for help, others wondering aloud where all the gold has gone. *QuestBusters* liked this game a lot, praising its humor and calling its parser "the smartest...of the year."

### Gnome Ranger

This appears to have been the last game in Level 9's collaboration with Rainbird, and it continues the trend begun with Knight Orc. It stars Ingrid Bottomlow, an accident-prone gnome, and takes place in a fantasy realm where humor and alchemy seem mixed in equal parts. North, for example, is spelled "gnorth" and the prompt is "What gnaw?" Not side-splitting stuff, but typical of the playful whimsy which makes the game so much fun. A reviewer in the British magazine *New Atari User* liked it even better than Knight Orc.

### Time and Magik

This is the last of Level 9's trilogies patched together from past releases, and the first brought out by Level 9's new distributor, Mandarin Software. The first episode, *The Lords of Time* (designed by Sue Gazzard), came out in 1983 and was intended to be the first in a series. However, the Austins weren't satisfied with Gazzard's subsequent work, and the series went no further.

Two years later, Level 9 brought out *Red Moon* to much acclaim, and in the following year its sequel, *The Price of Magik*. Both were designed by David Williamson, take place in the realm of Baskalos, and involve (respectively) the search for a red crystal and the defeat of an evil sorcerer. Unlike *The Lords of Time*, both had graphics.

After a bit of editing, these games were repackaged as *Time and Magik*, and distributed by Mandarin in the U.K. and Datasoft in America. Following the pattern established by Knight Orc, the Atari 8-bit version came without graphics (though this meant removing those originally present in *Red Moon* and *The Price of Magik*), while 16-bit versions had illustrations worthy of their capabilities. Again, *Time and Magic* elicited a glowing review in *QuestBusters*, and was declared one of the top five games in this genre.

### Lancelot

The second and final game released by Mandarin for Level 9, Lancelot continues the formula established by Knight Orc and Gnome Ranger—minus the humor. It is

based fairly closely on Malory's version of the Arthurian legends, though there are of course a certain number of alterations.

One of the most interesting aspects of this game is the way in which it was marketed. Purchasers of the game were given a chance to compete in a real-life grail quest, with the winner receiving a silver chalice worth 5000 pounds sterling. Despite this bold attempt at promotion, it seems Lancelot may not have been one of Level 9's more successful games. The review in *New Atari User* wasn't negative, but neither was it effusive. *QuestBusters*, however, was quite definite about its shortcomings.

### Ingrid's Back

With this game, Level 9 went back to distributing its own titles, and also returned to its successful formula of fantasy + humour. As the title suggests, it's a sequel to Gnome Ranger and stars Ingrid Bottomlow once again. The three episodes (*Little Moaning*, *Steamroller at Dawn*, and *Ridley's End*) chronicle Ingrid's fight to save her village, *Little Moaning*, from being razed by a land developer, Jasper Quickbuck. Part 2 parodies a famous scene from *Hitchhiker's Guide to the Galaxy*, and Part 3 finds Ingrid carrying her battle to Jasper Quickbuck's home, Ridley Manor.

### Scapeghost

This is the final text adventure produced by Level 9, and it has a twist which adventurers will appreciate: you begin the game already dead. Your task is to thwart the drug dealers who killed you, a challenge which at first seems formidable since, as a ghost, interacting with the physical world is a little difficult. According to the review in *New Atari User*, Scapeghost "oozes atmosphere and humour" and is "another winner." It was released in 1989.

### Other Games?

Yes, there's more. *Emerald Isle* is a text-only game released around the same time as *Red Moon* (1985), and was set in the Bermuda Triangle. It seems to have been intended as a beginner's level game.

Two others are *The Archers* and *The Secret Diary of Adrian Mole*. Both are illustrated but have no parser. On numerous occasions throughout the game, the player is presented with a choice of actions, and each decision becomes a branching point in the story. Both games were distributed by a company called Mosaic, but I'm unable to say exactly when.

There could well be other Level 9 adventures that I'm not aware of. Most of my information about the company and its products has been pieced together from reviews and articles in various magazines, including *ANTIC*, *QuestBusters*, *Computer Gaming World*, and the British magazines *Atari User* and *New Atari User*.

### The Best Part

If any of this has piqued your interest in Level 9, then the good news is that some of their games are still available, though you'll have to order them from England. The best way to find out about the availability of Level 9 software, as well as other great programs that never reached North America, is to subscribe to *New Atari User*. The magazine has just celebrated its 10th year of publication, and continues to be an extremely valuable resource for all dedicated 8-biters. Two-thirds of each issue is devoted to 8-bits, with type-in programs, useful articles, and best of



all, reviews of many sparkling NEW programs which continue to be released in Europe. Write to:

Page 6 Publishing  
P.O. Box 54  
Stafford ST16 1DR  
England

Subscription rates are reasonable- a one-year sub (6 issues) will cost you 21 pounds sterling (less if you opt for sea mail). VISA and MasterCard are accepted. *[Editor's Note: for potential USA subscribers, the currency exchange rate is roughly \$1.50 to the British pound.]*

*New Atari User* also has many back issues available at bargain prices. For text adventurers, I recommend issue #34 (July/August 1988), which is packed with information and features the work of two excellent reviewers, Garry Francis and John Sweeney. In #34 you'll find John Sweeney's seven-page interview with Level 9, and his reviews of Knight Orc, Gnome Ranger, and Time and Magik. From Garry Francis there's an extensive bibliography of adventures containing many titles you won't have heard of before, and six pages of hints for numerous games. The issue also has two type-in text adventures, Klepht's Castle and Demon Adventure. Finally, if you order the disk, you get two bonus adventures, Moonstone and Dedridge Castle.

(By the way, it was John Sweeney who eventually walked away with Lancelot's silver chalice.)

### Excel Update

If you want a break from text adventuring, try Lizard, the new shootemup just released in England. The graphics were done by Robert Stuart, who publishes Excel, the disk magazine mentioned last issue. Excel's prices were recently lowered to 2.50 pounds sterling per issue (current and back).

Till next time...



## BBS

### Express! Professional

You have tried the rest, now buy the Best!

Single Version.....	\$ 39.95
Multiline Version.....	\$ 89.95
Multiline Version & Hardware.....	\$235.00
Networking Modules.....	\$ 10.00
Xmodem Batch UL/DL Modules.....	\$ 20.00
Pro_Term Terminal Modules.....	\$ 20.00
Hardback & Restore H/D Util.....	\$ 20.00
D-K Utilities for Black Box.....	\$ 20.00
KPI Hard Drive Interface.....	\$149.95

Above items are for the Atari (r) 8-Bits  
Shipping/Handling for software..... \$ 4.00

K-Products  
P. O. Box 22122 AMF  
Salt Lake City, Ut. 84122  
(801)-967-7400

## Correspondent's Corner: Survivin' On The Island

Don Lebow, AC Correspondent, Hawaii

### Isolated in Paradise?

When it comes to feeling isolated with an Atari Classic in a PC world, I sometimes feel more isolated than most. After all, I live on a very small rock in a very large pond called the Pacific Ocean. From my perch here on the tropical island of Maui in the Hawaiian Islands, it's 2500 miles to anywhere. Even visiting Honolulu requires a somewhat pricey plane flight. There are compensations, mind you <ahem!>, but as far as things fuji, it would seem I'm pretty much on my own, no?

Not really. As many have discovered, the key to an ongoing relationship with your "orphan" computer is networking. No, no... I don't mean go out and look for an Ethernet card for your XE! I mean logging on to the worldwide network of 8-bit users. And they are out there, believe me. That means using a modem to tap into the bitstream. Talk about your wonders of modern science! With modems so inexpensive these days it's easy to take telecommunications for granted (I confess to that), but if you think about it, it's a pretty amazing phenomenon. And it's saved my bacon innumerable times.

### Long-Distance Repairs—and Friends!

For instance, there was the time the keyboard on my XE went down. Not being a techie (I'm one of those who can never remember which end of the soldering iron one picks up), I was leery of trying to replace it. Hey, they said, anyone can do it. Not ME!, I replied. Finally Hardware Wizard Bob Woolley stepped in and told me not to worry. He would provide step by step instructions. Here's what he wrote:

1. Pick up screwdriver.
2. Remove old keyboard.
3. Put in new keyboard.

And by golly, it worked! Sometimes it takes a long distance buddy to convince you that you're capable of more than you thought.

And that's the best part, of course. You people out there. In 8 years of telecomputing, I've met a lot of folks. Nice folks. It's hard to explain to someone unfamiliar with telecom how you can have a "good friend" whom you've never actually "met." But the phenomenon is very real, and it's one of the best things that's happened to me since I got into personal computing.

### Staying Connected

It's not really necessary for you to log onto one of the commercial services to start getting in touch (though if you drop by ATARI8 on CompuServe, be sure to say hello!). A local BBS is just as good a place to start as any, and there are more than you might think. If a BBS is hooked into a network like FIDONET, your reach is that much wider. If you have Internet access, consider checking out the very active 8-Bit discussion group.

It goes without saying that being so far away from everything also means I have to rely totally on Mail Order (not unusual nowadays.) We're fortunate in having some dedicated vendors who continue to support the 8-bit, like Bob Puff at CSS (yup, that's an unsolicited plug!). It's amazing what stuff you can find out there. I never thought I'd see the day when I'd be running both a hard drive and a high density 3 1/2" floppy on my little XL, but there it is, right in front of me. Who says there's nothing new? Not to mention, it sure is nice to be able to call one of the vendors in AC, say something along the lines of, "I need an XL power supply", and actually have them know what you mean! Don't try this at your local Radio Shack... <heh>.

I do practice one basic survival technique: I hoard equipment. Never know when something might go down, and you'll need a backup, while your regular is taking a long long plane ride to somewhere. The alternative is too grim to contemplate... Aloha!



## Exploring The Wild FONTier

by David Richardson  
AC Staff Columnist

Reprinted by kind permission of  
the Lawrence Atari Computer Club  
of Lawrence, Kansas

This column is not the result of the desk top publishing of Unicorn Publications. It was printed entirely by a program called Daisy Dot III, a print processor. The printer used was a Star NX-1000, and the DOS was SpartaDos 3.2d. A camera-ready copy was sent to the publishers and they then inserted it directly into the magazine. DD3 is a shareware program. See the information at the end of this article if you are interested in acquiring DD3.

Here we go again with yet another trek into the "Wild FONTier". On this trip, let's talk about double-column printing.

DD3 doesn't have true double-column printing, in that it doesn't print both columns at once. To print double columns, print the left column first, roll back the paper, and then print the right column. Also, when you print that left column, use the print pages option from the DD3 menu and set it to say that you want to print only the ODD pages. Once the ODD pages, or the left column, is done, roll back the paper, change the margins that correspond to the right column, and then print the EVEN pages.

The most important thing to remember is that when you change the margins for the right column, make absolutely sure that the total width of the right column is EXACTLY the same as the left column. If they're not EXACTLY the same, then it's almost guaranteed that the text from one column to the next will not match up. This isn't quite as hard to accomplish as it sounds. There are certain things you can do to make this task easier.

### The Layout

The first thing I do is determine how wide I want each column to be, in inches. For lack of a better point of reference, let's use the column you are now reading. Each column is 3 and three-eighths inches wide, with one-quarter of an inch between them, for a total printed page width of 7 inches. Knowing these widths, I can then use the left and right margin formatting commands to make the changes.

You'll soon realize that in order to print both columns, you'll have to print the first column (ODD pages only), exit DD3 and go to the word processor, change the margins of the text file, re-save the file with the new margins, and then re-load DD3, remembering to roll back the paper. You then re-print the text file again (EVEN pages only). This, of course, is both confusing and time consuming. There is a way to accomplish the goal in a more practical manner.

### Don't Get Ahead Of Yourself

One of the easiest ways to simplify things is to write the entire text file as if you'll print the left column only, using the left and right margin settings for the left column.

I CANNOT STRESS THIS POINT ENOUGH. Don't even bother with trying to figure out how to do the right column until you've done all of your proofreading, rewriting, editing, etc. The reason I say this is, imagine you don't have the finalized version and you save both the left and right columns. Then, you find a mistake or decide to re-write it. Now, both files are wrong, and you'll have to re-do both. It's much easier to get all of this out of the way, and then all you have to worry about is the margin change. You now go to the next step.

You now save the text file for the left column only to the disk. If you wanted, you could then load up DD3 and print out every page, the result being that all the information would print out, but only the left half of the paper would have any text. Obviously, this is not what you want. Rather than exit the word processor and then come back to it, simply make the margin changes in the already existing text file, but save it with another name. I use the same name for both files, but I use the extension \*.L for the left, and the extension \*.R for the right. Both files are EXACTLY the same except for the margin changes. I now have two files on the disk. I now go to DD3, print the \*.L text file first, ODD pages only. Once this is done, roll back the paper. Load in the \*.R text file and print the EVEN pages only.

One thing to remember using this method is that if you make margin changes within the text file other than the initial ones at the beginning, you must go through the ENTIRE file and change EVERY margin command, for both left and right margins. I avoid this problem by writing my article without ANY margin commands except for the ones at the beginning of the file. This way I know that the second column will turn out correctly.

The next thing to do is to decide what the margin values should be for the respective columns, and how to set the DD3 formatting commands for both text files. DD3 does its calculations for 8" wide paper, not 8.5" wide. In the case of the column you are now reading, for the left margin of the left column, a half-inch left margin translates into the DD3 command of \XL020. As for the right margin for the left column, I figure the distance from the right side of the page. I added the space between the two columns (one-fourth of an inch), the width of the right column (three and three-eighths), and the width of the right margin of the right column (half-inch). These add up to four and one-eighth of an inch. Convert it to decimal (4.125") Since a horizontal value of 40 equals 1", 40 multiplied by 4.125" figures out to 165. Thus, the command for the right margin of the left column is \XR165. The text file for the left column should have as its margins: \XL020\XR165. I enter these settings and call the file \*.L. For the settings for the right column, simply swap the values: \XL165\XR020. I enter these values and save it as \*.R. Doing this in this manner guarantees symmetrical columns. I now have two identical files, except for the margins.

## Print Preview

At this point I should tell you about a feature of DD3 that you probably don't know about unless you read the documentation that comes with it. This is the "print preview" option. Some word processors also have a print preview option. What it does is it tells you how long your article or text file would be if you actually printed it out. This is handy if you want an exact number of pages. If you use TextPro, for instance, you do this by simply placing the cursor at any point in the document and then hit CONTROL-W. Think of this as the "Where?" command. TextPro will then tell you "where" in the document the cursor would be if it were printed out. It would be in the form of, "Page 3 of 6, line 20 of 60". You can then use this method to see where your page breaks are. To see how long your document is, simply place the cursor at the end of the file, and hit CONTROL-W. If it's too long or too short, you can change the document accordingly.

TextPro even has a command to tell you how many words are in the document. AtariWriter has a graphically represented print preview. While in the edit mode in AtariWriter, you hit OPTION-P, and you'll get a page by page "picture" of the document. You can scroll the window around using the arrow keys and see exactly how the pages will actually look like if printed out. The print preview option of DD3 isn't quite as informative as these two examples, but it still has its uses.

You access the print preview from the DD3 main menu. Normally once you have entered the font name and the text file name, you then hit START and it proceeds to print. If you don't want to start with page 1, you can change it where it says "Begin Page". It normally starts at page 1. This can be any number from 1 to 255. To access the print preview, change this value to any number that is larger than the actual document. Of course, there is no way to know this. However, it is doubtful that you will have a document that is 255 pages long. If you change this to 255 and then hit START, the screen will go blank, as DD3 always does, and it will calculate each page, one at a time.

One thing to remember is that your printer must be on, because DD3 is expecting to actually print out a document. It checks to see if the printer is on, and if it is not, then it will give you an error message. If it goes through the entire document and has not reached page 255, the screen will come back on, and at the bottom it will tell you how many pages it got to.

For my article, I usually try and make it about four pages, or columns, and I prefer that it use the full length of the column with no wasted space. Unlike TextPro or AtariWriter, DD3's print preview only tells the number of pages, and you have no idea how many lines are on that last page. It could be full, or it may have only 1 line in it. You could print it out, of course, but then why bother with the print preview. The whole reason is to avoid unnecessary printouts.

Usually, I'm pretty good at guessing how long an article will be, and most of the time the print preview is either 4 or 5 pages. To get it exactly as I wanted, I used to print it out once first, and would then see how far off it was from 4 pages. Then I found a way to do it differently. If the print preview says 4, then it is either exactly 4 (not likely), or it is not quite 4.

I load up the word processor, usually TextPro. If DD3 said it did not go over 4 pages, I go to the beginning of the

text file and change the original line spacing command, usually \XV02, and change it to \XV03, making it have a slightly larger line spacing. I then save the text file to the disk, and re-load DD3. Again I use the print preview just like before, but now the article should be longer, since the line spacing is larger. Usually the print preview now shows that it is 5 pages, which of course is too long. Now I know that the ideal situation is somewhere in between. I then exit DD3 and go back to the word processor and change the line spacing command back to \XV02, which now makes my article too short again. This time, however, I move the cursor somewhere towards the middle of the text file and add the command \XV03. What this does is, the printout up to this point will have the line spacing of 2, and from this point on will have the line spacing of 3. Since this is only a difference of 1/72 of an inch, it will not be noticeable, at least not unless you are looking for it.

You must remember that this command is only legal in the first line of a file, before the first character to be printed as text on that line, or at the beginning of a new line of a file, before any characters to be printed on that line. Thus, it can be placed anywhere after a return, or the first thing in a paragraph. If you do not do this, it will not be recognized by DD3 as a line spacing command change.

I then save the text file again and go back to DD3 and use the print preview the same way as before. If it says 5 pages again, that means that the larger line spacing command, \XV03, was placed too high in the file. The closer the larger line spacing command is to the beginning of the file, the larger the overall printed file will be. If it says that it is 4 pages, that means that the larger command was placed too low in the file. So, if it says 5, I delete the command where I put it and I move it further down the file and re-save it. I then load up DD3 and re-do the print preview. By going back and forth I can narrow down the point at which the line spacing command should go. Once this is determined, save the file for the last time, and then save another one with the margin changes for the right column. This is the only way of having a print preview for MyDos. There is a utility for use with SpartaDos that will give you a graphical representation of a page. More about that in another column. I'd prefer to get it working with MyDos also before I get into it here.

Well, that's all there is to it. I realize all of this might sound a bit complicated, but it really isn't. Like anything else, once you get into it, you'll see that it is pretty easy to do, it just takes a little bit of forethought to get everything just right. And once you see the results of your efforts, I'm sure you'll agree that it's all worth it.

As always, if you have any questions, I'll be glad to help.

David Richardson  
P.O. Box 746  
Lawrence, KS 66044  
Phone: 913-843-5213

Fidonet Address: DAVID RICHARDSON on 1:299/102

Internet Address: david.richardson@f102.n299.z1.fidonet.org



# Bring your Eight up to Date

## with power products from **Computer Software Services**

### The Black Box

The Black Box is an add-on board for the Atari 600XL (upgraded), 800XL, and 130XE computers. The Black Box provides many unique and useful functions.

**The RS-232 port** emulates the Atari 850 interface very closely, but goes beyond by providing 19,200 baud capability. The Black Box is the only interface to support **hardware flow control**. This enables owners of high speed modems (4800 baud and up) to use their modem at full speed, and not worry about data loss due to a slow BBS or terminal program.

**The parallel printer port** interfaces to most all Centronics-type printers. You may assign the printer number and line feed options from within the Black Box's configuration menu. The Black Box allows **buffering** of your data, either using the extra 64K in your 130XE or the optional 64K RAM in the Black Box itself.

**The hard disk port** was the real reason for the design of the Black Box. You may connect most any hard disk controller that is SASI or SCSI compatible, or drives with embedded SCSI controllers. It is totally compatible with the current versions of MYDOS and SpartaDOS. The Black Box allows you to have up to 96 partitions with names, and set any partition as any drive (D1: through D9:), allowing you to place unprotected single density boot programs on your hard disk. You may also write protect ALL of your hard disks with the flip of a switch.

The Black Box also provides support for users who have previously used an ICD MIO to store data on a hard disk. You can have compatibility with an MIO formatted hard disk by simply setting a dip switch.

**The 6502 monitor** is very handy for machine language programmers. How often have you wondered where your program was, or what caused an apparent "lockup"? Entering the monitor will show you all the processor registers, and display the disassembly of the instruction it was about to execute when you pressed the button.

**A printer dump** of your current screen may be done at any time by pressing one of the buttons on the Black Box. A switch allows you to choose either text or graphics dump. (The graphics dump is only available for dot matrix printers capable of graphics.)

CSS sells a variety of accessories for your Black Box. The **Black Box Case** is a durable black plastic housing for your Black Box which sells for \$39.95. **Modem and printer cables** are available for \$9.95 each.

The Black Box sells for \$199.95 plus \$8 shipping and handling. The Black Box with a 64K printer spooler sells for \$249.95 plus \$8 shipping and handling.

### The Floppy Board

Our latest and greatest product! The Floppy Board is an add-on expansion board for the Black Box interface. It allows the user to add up to 4 inexpensive, 'standard' floppy drive mechanisms. The Floppy Board supports almost all floppy drive configurations, including 360K, 720K, 1.2MB, and 1.44MB. Built-in the Floppy Board are the Black Box Enhancer and a version of our Super Archiver.

Disks formatted on the Floppy Board are accessed at parallel bus speeds, providing a **substantial** performance increase over the standard serial Atari floppy drive. In fact, Floppy Board drive access is much closer to hard drive speeds than to standard serial drives! Included with the Floppy Board is our **IBM/ST Disk Transfer Utility** program, which allows you to both read and write IBM or ST disks. This makes the Floppy Board the ideal method for porting files to and from your PC or ST!

Also available for your Floppy Board is our 1.44MB drive kit. The kit includes a high-quality 1.44MB drive mechanism, power supply, and floppy drive cable, and sells for only \$79.95 plus \$8 shipping and handling. The Floppy Board sells for \$149.95 plus \$5 shipping and handling. CSS also sells power supplies, floppy drive mechs, and custom floppy drive cables. Call for pricing.

### Black Box/Floppy Board Special!

For a limited time, CSS is offering special pricing when you buy both the Black Box and Floppy Board. When purchased separately, you would spend \$349.90 to get these items. Our special pricing lets you have these products for only \$329, a savings of over \$20!

### 8-bit Repairs

Having a problem with your 8-bit equipment? CSS repairs all kinds of Atari products, from computers to disk drives. Call us to arrange shipment of your items.

If you have two of the same product and both need repair, take advantage of our **two for one** repair deal. Send both items to us, and we will fix one of them **FREE** and keep the other one for parts. Again, call us to arrange shipment.

### The Black Box Enhancer

A must for all Black Box owners! The Black Box Enhancer is a plug-in module for your Black Box, enhancing the printer functions and adding an instantly available, full featured **sector editor**!

The built-in **screen dump to printer function** will now render a hardcopy with a 16 shade grey scale representation of the colors. A special graphics printer handler is built-in, which allows any output directed to P9: to be printed on your graphics compatible printer with the same character font used on your computer. This means all the Atari special control and graphics characters will be printed, along with inverse.

The built-in Task Master sector editor is the most powerful editor for the 8-bit. It contains a sector copier featuring multiple copies, automatic formatting, and uses all available memory for fast disk duplication. The Task Master is not limited to only floppy disks. It can handle up to 16 megabyte hard disk partitions (even in the sector copier mode)! The Task Master provides full DOS support for MYDOS, SpartaDOS, and Atari DOS derivatives. Subdirectories are fully supported! You may link through individual files by simply moving through the directory and highlighting the file you wish to edit. 16-bit and sector map linking are supported for hard disks, and 11-bit linking for floppies. It is ideal for quickly editing files and repairing damaged directories.

The Black Box Enhancer sells for \$49.95 plus \$5 shipping and handling.

### Ordering Information

Call CSS at (716) 429-5639 between 10am and 5pm Eastern Time to order, or send your order to the address below. CSS accepts payments in cash, money order, Visa, MasterCard, or COD. Personal checks are also accepted, but shipment of your product is delayed until the check has cleared. Foreign orders must be paid in US funds.

Call or write for our free catalog of products. CSS also has a free catalog of pre-owned software, Atari related books, and owner's manuals for most hardware products.

### Other Products

Bit-Writer (for 1050 drives).....	\$ 79.95
Black Patch (for all computers).....	\$ 29.95
Disk Cracker (for all computers).....	\$ 19.95
Hard disk controllers (for all computers).....	CALL
Hard disk systems (600/800XL and 130XE computers).....	CALL
HD Pro Backup (for Black Box owners).....	\$ 49.95
Heavy Duty power supplies (600/800XL and 65/130XE computers).....	\$ 17.95
Multiplexer including master & 2 slaves (for all computers).....	\$199.95
Multiplexer extra slave (for all computers).....	\$ 89.95
Power Plus memory upgrade (800XL and 130XE computers).....	\$ 99.95
Quintopus (for all computers).....	\$ 39.95
Quintopus with switchable ports (for all computers).....	\$ 59.95
Richman's Word Processor (for all computers).....	\$ 39.95
Silencer (for 1050 drives).....	\$ 19.95
SIO Power Booster (for all computers).....	\$ 39.95
Super Archiver (for 1050 drives).....	\$ 69.95
Super Archiver II (for 1050 drives).....	\$ 99.95
Upgrade from SA to SA II.....	\$ 29.95
Super-E EPROM Burner (for XL and XE computers).....	\$169.95
Gang Super-E EPROM Burner (for XL and XE computers).....	\$199.95
Super Pill (for all computers).....	\$ 24.95
Ultra Speed Plus OS (for XL and XE computers).....	\$ 69.95
XF Dual Drive Upgrade (for XF551 drives).....	\$139.95
XF Dual Drive without drive mech (for XF551 drives).....	\$ 79.95
XF Enhancer (for XF551 drives).....	\$ 34.95
XF Single Drive Upgrade (for XF551 drives).....	\$ 99.95
XF Single Drive without drive mech (for XF551 drives).....	\$ 59.95
XF Update (for XF551 drives).....	\$ 39.95

**Computer Software Services, Inc.**

**PO Box 17660**

**Rochester, New York 14617**

**Phone: (716)429-5639, 10am-5pm Eastern**

**Fax: (716)247-7158**

**BBS: (716)247-7157, 300-9600 bps**

# Cheep Talk Type 'n' Talk

LOUIS MARCOTTE, AC CONTRIBUTING AUTHOR

## Lee Brilliant's "Cheep Talk"

In the April 1985 issue of *ANALOG Computing* magazine, Dr. Lee Brilliant presented plans for an inexpensive speech synthesizer called "Cheep Talk" that could be connected to any 8-bit Atari computer through the joystick ports. I immediately rushed to the nearest Radio Shack store to purchase the parts and set to work with my soldering iron. Upon completion, I plugged it into my 130XE and ran the test program included with the article. I was pleasantly surprised to find it worked on the first try, and the speech quality was pretty good.

## Type 'n' Talk: the T: Driver

Dr. Brilliant had described how to manually convert English phrases into elements of speech called allophones that can be sent to Cheep Talk using a simple BASIC program to produce speech. This approach produces excellent results but is tedious and time-consuming, requiring all speech to be hardcoded within the program.

A more versatile approach is to have the computer translate text into speech allophones in real time, allowing any English sentence to be spoken. Cheep Talk Type 'n' Talk uses a custom text-to-speech conversion algorithm to provide an unlimited vocabulary and an easy to use interface for Dr. Brilliant's speech synthesizer. A general purpose Cheep Talk device driver, T:, is also provided which allows easy access to the text-to-speech conversion process from programming languages such as BASIC or assembly language, or from a Disk Operating System (DOS).

## Getting Started

Cheep Talk Type 'n' Talk and the T: device driver are written entirely in machine-language and will appear on AC's June, 1993 Software Disk as the following files:

Cheep Talk Type 'n' Talk	TYPETALK.COM
T: Device Driver	T.COM

The above files are also available from the online services (GEnie, CompuServe) in ARC'ed form.

Connect the speech synthesizer unit to joystick ports one and two, and load either TYPETALK.COM or T.COM using the Binary Load option of your DOS.

## Using the Programs

The heart of these programs is a custom text-to-speech conversion algorithm which regulates all speech activity. All ASCII text directed to this algorithm is converted to speech output.

Cheep Talk Type 'n' Talk accepts English text sentences from the keyboard and outputs them to the speech synthesizer via the text-to-speech conversion process. Just type a sentence at the prompt, press RETURN, and the sentence will be spoken. Full screen editing is supported, allowing previous sentences to be re-spoken simply by re-positioning the cursor anywhere within the desired sentence and pressing the RETURN key. Sentences can also be re-edited and spoken in a similar manner. Non-alphabetic characters are ignored, except for the comma, space, or period characters, which produce appropriate pauses when used between words or sentences.

The T: device driver is loaded into memory when the T.COM file is executed. This file can be renamed AUTORUN.SYS so the driver will be loaded upon bootup. The T: device is protected from system RESETs and will remain installed until the computer is turned off. Programming the T: device is done using the Central I/O (CIO) system, either from assembly language or a high order language (HOL) such as BASIC.

The following example in BASIC that will open a channel to the T: device, send a test message to be spoken by the speech synthesizer, and then close the T: device channel.

```
10 CLOSE #1
20 OPEN #1,0,0,"T:"
30 PRINT #1,"TESTING TESTING TESTING"
40 CLOSE #1
```

I recommend you only use channels 1 through 7 since channel 0 is used by the operating system for the screen editor (I used channel 1 in this example). A similar approach can be used for other HOLs or assembly language.

## Speech Quality

Since a speech algorithm is used to translate English phrases into speech output, any word or combination of letters can be spoken, even if the phrase is gibberish. This provides an unlimited vocabulary. However, the quality of the spoken words is not as good as if a speech dictionary were used. I've optimized the algorithm so that most

commonly used words are understandable. The English language is very complex, however, and the program will occasionally mis-pronounce words. In these cases, intentional mis-spelling of the word will almost always produce the correct pronunciation. For example, it's impossible for the program to distinguish between a word that's spelled the same but pronounced differently, such as the word 'read' in the following sentences:

I have read this article.

I will read this article.

Using 'red' instead of 'read' in the first sentence will produce the desired results, though perfectionists might find this annoying. Similarly, words such as 'lose' can be spelled 'looz' to achieve the desired pronunciation.

### Applications

Applications for the T: device driver are limited only by your imagination. Possible applications are burglar alarm systems, home control systems, games, educational software or wherever controllable synthesized speech output is desired.

Cheep Talk Type 'n Talk can be used to quickly prototype speech text to be inserted into a program that uses the T: device. Its editing facilities allow you to modify text until the desired pronunciation is obtained. Or you can simply use it to amaze your friends with your talking Atari!

### Programming Note

Cheep Talk Type 'n Talk was developed using the Six Forks Assembler/Linker package from Six Forks Software. This linker-based assembler facilitates the design and development of complex software by allowing the code to be broken down into functional units. Each unit has a single function that's easily designed, coded and tested. The units can then be integrated together in order of increasing complexity until the entire program is functional. This drastically reduces program development time and results in code that's easy to understand and maintain.

### Conclusion

I hope many uses are found for this nifty Atari 8-bit capability. If you haven't already built the speech synthesizer, this is a good reason to do so. Get issue #29 of ANALOG Computing and follow the directions in the article. If you aren't electronically oriented, perhaps you can ask a friend who is to help. One possible problem will be finding the SPO256-AL2 speech synthesizer chip, since Radio Shack no longer sells them. You may have to hunt around other electronic supply sources to find it, or you may have to purchase a used Cheep Talk device. Good luck!

If you have comments about the programs or would like to describe your use of the programs,

please drop me a line care of *Atari Classics* magazine. I hope you have as much fun using these programs as I had writing them.

### About the author:

Lou Marcotte is a Senior Design and Development Engineer at Raytheon Company, where he develops software to control military and civilian radar systems. This is his first published article.

*Editors's Comments: Finding the SPO256-AL2 chip will indeed be a problem. I looked through my pile of electronic component catalogs and finally located the chip in the 1991 Active Electronics catalog for \$9.95. That was the most recent edition I have; however Lou indicates it is still listed for the same price in in a more recent edition of the Active catalog. Active is located at 11 Cummings Park, Woburn, MA 01801 USA, phone 617-932-4616.*

*I built the Cheep Talk speech module myself but hadn't used it for quite a while; like Lou, I found programming the individual phonemes with BASIC too exhausting. The Type 'n' Talk programs are a major improvement over Lee Brilliant's original software, in that you get a fast aural feedback. Aural memory fades rather quickly, and allophone programming takes so long you forget what the previous "take" sounded like while you're working on the next improvement.*

*In trying out the programs I found a few nits to pick. I found myself wishing an XEP-80 driver had been included, just so I could keep more text on the screen to prevent my repeating pronunciation trials. The algorithm Lou used could stand some improvement. The T: driver handles consonants pretty well, but it gets confused on vowels and vowel combinations. I tried a few tongue-twisters on it for fun. "Peter Piper picked a peck of pickled peppers" had to be typed as "PEETER PI PER PICKED A PECK OF PICKLD PEPPERS". When spelled correctly, Peter Piper came out sounding like "Petter Pipper". I also tried "How much wood would a woodchuck chuck if a woodchuck could chuck wood?", and that one took some effort. Eventually I found that "HAWO MUCH WIOD WIOD A WIODCHUCK CHUCK IF A WIODCHUCK KIOD CHUCK WIOD?" sounded more normal on the Cheep Talk. There's a problem with long "o" sounds; "mud" comes out sounding like "mood".*

*Even so, it would have taken me hours or days to get Cheep Talk to pronounce these sentences with phoneme programming. After awhile I began to get the hang of it, but only after I started making some very weird sounds of my own, listening to them, then trying to develop a phonetic spelling. I'd liken the experience to teaching a speech-disabled child how to talk by listening to the sound of your own speech. It's a faculty we take for granted; I found the experience enlightening. - BLP*





# Moonlight Workshop

JEFF McWILLIAMS, AC Staff Columnist

## The XL/XE Guide to Expanded Memory



### Most Popular Upgrade?

What do you suppose are the two most popular upgrades we 8-bit users purchase these days? Most likely a higher capacity floppy or hard drive is one, and a memory upgrade is the other. That shouldn't surprise anyone; trends in our community often follow trends in the PC world, and everybody knows that for PC users the more RAM and storage space you have, the better. While 8-bit power users can look to 256 or 512K memory upgrades, and 20 megabyte hard drives, PC users are looking at 4-16 megabytes of memory and hundreds of megs of hard drive space.

Back when I purchased my first 800XL, the 64K of RAM that it contained was more than enough for me. At that time there weren't more than a few programs that used more than 64K on an upgraded 800. People happily used programs like AtariWriter and Visicalc with 90K or 180K floppy drives and seldom complained. But as time passed, people's needs and wants have changed. We saw the sophisticated new software coming onto the PC market and wanted similar features in programs for our trusty 8-bits. I don't think we ever expect to run AutoCAD or Word for Windows; programs like that require massive amounts of memory and CPU horsepower. What we did want were programs like Paperclip: a word processor supporting expanded memory, memory resident spell checker, indexes, tables of contents, and lots of other bells and whistles.

At the same time we were lusting after a new generation of applications, RAM chips became really affordable. Even tightfisted 8-biters could afford the price of 256K DRAMS. It was a natural course of evolution for memory upgrades to become popular so that more sophisticated programs could be written to use all that RAM.

### RAM Limit: A Sticky Wicket

There's a small technical problem however. While the Intel microprocessor in a PC accesses upwards of 640K of RAM, the 6502 microprocessor in our Ataris can only access 64K. This isn't due to faulty engineering at Atari. In 1979 when the 800 came out, the 6502 gave the most bang for the buck. The Intel based PC came out later and used a newer CPU that supported more RAM and could be expanded as the need arose. Meanwhile, 8-bit Atari computers were already operating with the maximum memory a 6502 could handle. It would be easier to squeeze water from a stone than get a 6502 to access more than 64K of RAM. What was to be done? How would the hardware people meet the user community's insatiable desire for MORE RAM?

If you've been an 8-bit user for any length of time, you know us die-hards never say never. I like to think that nothing is impossible on an 8-bit, it's just more challenging. Such was the case with breaking the 64K memory

barrier. Once they put their mind to it, entrepreneurial hardware hackers solved the memory problem and started churning out 256K RAM upgrades left and right. By about 1986 or so, everyone and their brother seemed to be either buying their own memory upgrade or installing one for somebody else. I remember reading how one guy took his 800XL to a computer show and had 256K installed while he waited. Now that's what I call service! Then Atari Corp. wised up and put out an expanded RAM machine, the 130XE with 128K. Its method of expanded RAM set the standard of compatibility for all the homebrew upgrades on the market.

### More RAM Than Ever

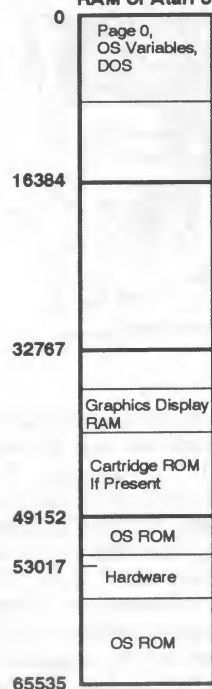
So here we are today, with RAM upgrades as popular as ever. Not as many companies sell them, though—a reflection of the shrinking 8-bit community, and not a decreasing need for more memory. If we ever hope to make further progress, to write 24-pin printer versions of Daisy Dot III or high resolution Schematic and generic CAD applications, we'll need nice big memory upgrades for them to be useable. This translates to a continued demand for RAM upgrades. Despite the 8-bit glory days having passed into the sunset, there are more programs than ever that utilize expanded RAM in some way.

Although RAM upgrades are an old idea, I still get bombarded with questions from people seeking information about them. Not all Atari users are seasoned veterans. Our community is still acquiring new users, people picking up used equipment at garage sales or inheriting it from other family members. This is a good time to thoroughly explore memory upgrades, how they're done, and what software is out there to use the RAM once you've got it. So read on! This is an enormous topic, and it might take me several installments to cover the salient points. I hope to start a regularly updated database of various expanded RAM applications to provide information I may miss the first time around. I'll need your help on that, so if you know of a program or utility that uses expanded RAM, please write to me *c/o AC!*

### Understanding Banked RAM

Visualize memory in your mind's eye as a long vertical strip of paper with one piece of data being written per line on the strip. The paper has a finite width and length. Each line on the strip is numbered. If the first line is numbered 0 and the last is numbered 65535, then you can store 65536 numbers on the paper (or 64K in computerese). What happens if you want to store more than 64K? You can't make the strip of paper longer because you're limited by the design of the microprocessor.

### Simplified Layout of 64K RAM of Atari 8 Bit



Now suppose you make a horizontal strip of paper you can pass through one section of the vertical strip. By sliding the horizontal strip of paper left or right, a certain portion of the horizontal strip will become part of the vertical one. (See diagram below.) In this way, you would get more surface area of paper to store numbers on. Or in our case, more accessible memory. In essence, this is how memory upgrades for the Atari 8-bit work. There's a memory location called Port B located at 54017 (or \$D301 in hexadecimal notation.) By writing a number to this port, a chip inside the computer will set some of its pins to high or low voltage levels. When these pins are wired up to additional circuitry, specifically a RAM upgrade, they control the "sliding" left or right of the horizontal strip of paper.

The horizontal strip slices through the vertical one starting at location 16384 and ending at 32767. Thus the horizontal strip is 16K bytes tall. As you slide the horizontal strip left or right, 16K of data from the horizontal strip appears as part of the vertical strip and represents the memory "seen" by the microprocessor. The 16K area the horizontal strip moves through is sometimes called a window. Each time you slide the horizontal strip of paper left or right a new piece of the horizontal strip, 16K large, appears in the window. These 16K pieces are called banks. This is why this memory upgrade method is called *banked* RAM.

On the 130XE, there are five programmable settings at the \$D301 location. These settings correspond to which bank of RAM appears in that 16K window. You can have one of four external banks appear, or no bank appear. No bank would mean the original piece of the vertical strip resides in the 16K window. The 64K from the vertical strip plus the four 16K expanded banks equals 128K of RAM.

### Homebrews: With or Without ANTIC

The first major memory upgrade for the XL/XE machines was "The Quarter Meg 800XL" by Claus Bucholz. It appeared in the September 1985 issue of *BYTE* magazine, and it gave XL owners 256K of RAM. Claus had actually completed the design in December of 1984. His original design differed from the 130XE; it used 32K banks instead of 16K banks. Once the 130XE was released in early 1985, Claus promptly modified his design for compatibility with Atari's expanded memory design. Under this final standardized design, you divide up the 256K total RAM into 192K of external RAM (twelve 16K banks on the horizontal strip) plus the original 64K of RAM on the vertical strip in our paper strip model.

Claus' revised 256K upgrade works pretty much like the 130XE, but there's still one slight difference between the two. Like many other commercial and public domain memory upgrades created since then, the Bucholz upgrade doesn't fully support ANTIC banking. Oh gosh,

what's *that*?

ANTIC banking is a special mode that determines whether or not the video processor (ANTIC chip) also sees the expanded memory. ANTIC banking to date is used in only a few programs. Trying to explain ANTIC banking will make things needlessly complicated for you beginners. It's an issue that

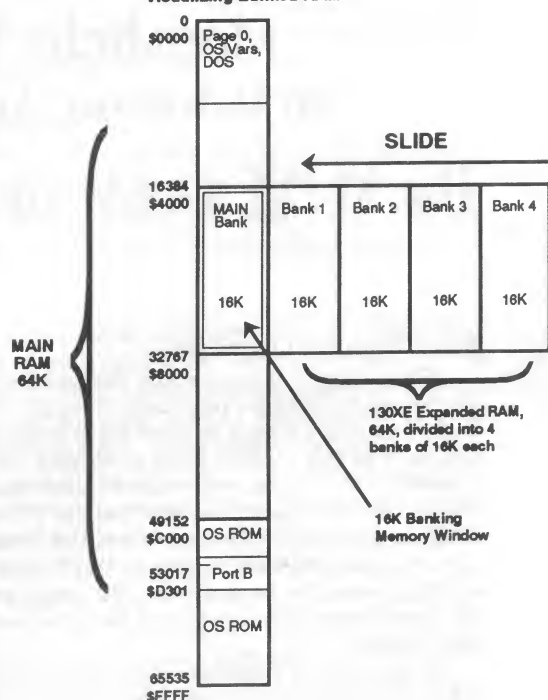
can be ignored in 95% of the situations where expanded memory is used. I'll come around to the concept of ANTIC banking again in a future installment, but for now I'm content to leave you with an awareness of its existence.

### RAM To The Max

Just how big an upgrade can you put in the computer? At present, the main limitation is the number of free bits in the Port B register at \$D301. For x number of bits you have free in that register, you can get  $2^x$  number of banks. There's an upgrade kit from Newell Industries that manages to take advantage of 6 of the bits in Port B. That yields 1024K of expanded memory; this is the upgrade I have in my 800XL. That's enough to satisfy even a hardcore power user like me! (Well, at least for a little while.) I'd like to experiment with adding a 4096K SIMM module to my 1200XL <grin!>, which for all practical purposes is the outer limit of expanded RAM in our machines. Even 1024K is pushing the limits of the Port B register. With the Newell 1-meg upgrade installed, you lose software control of the internal BASIC ROM chip plus a few other things you have to watch out for. Right now I just want you to get a good grasp of the fundamentals of how expanded RAM works so we can begin surveying applications that will use all that delicious extra memory.

If you can go through the information above and come out with a good idea of what's going on inside an expanded computer, you've got what it takes to move on. This is all the average user needs to know about HOW the memory upgrades work. The most important thing to remember is that the extra RAM is limited to appearing in 16K banks starting at \$4000. This is important because it can affect how the expanded memory is used by various software applications. We'll explore this next time and start our expanded memory software survey beginning with word processors. Until then, if you still have difficulty understanding banked RAM, make your own vertical strip/horizontal strip model like the one in the diagram. (If someone asks, say it's a craft project for your child!) See yah next time!

### Visualizing Banked RAM



# B&C COMPUTERVERSIONS

2730 Scott Boulevard - Santa Clara, CA 95050

Store Hours Tue - Fri 10am - 6pm  
Sat - 10am - 5pm Closed Sun - Mon

## 408-986-9960

### STE & MEGA COMPUTERS!!

#### 1040 STE

1 meg \$425  
2 meg \$495  
4 meg \$555  
Great Family Computer!

#### MEGA ST LOWEST PRICES EVER!!

2 meg \$495  
4 meg \$555  
The Original MEGA

#### TT030

**\$1295!!**  
2 meg ST RAM \$175  
4 meg FAST RAM \$299

### SOFTWARE SPECIALS FOR THE ST

limited quantities!!

**\$9.95**



Hundreds of other titles too!

POLICE QUEST II COLONELS BEQUEST  
CAMELOT CODENAME: ICEMAN  
MANHUNTER SPACE QUEST I  
MANHUNTER II SPACE QUEST II  
BLACKCAULDRON DE JA VUE II

HOYLE BOOK OF GAMES I  
HOYLE BOOK OF GAMES II  
MIXED UP MOTHER GOOSE  
SLEEPING GODS LIE  
GOLD RUSH  
GUNSHIP  
WEIRD DREAMS  
NETHERWORLD  
SILENT SERVICE



### RECONDITIONED MERCHANDISE

400 Computer 16K with BASIC \$24.95  
800 Computer 48K/Star Raiders \$59.95  
600XL computer with 16K \$39.95  
600XL computer with 64 K \$59.95  
810 disk drive \$100  
1050 Disk Drive - dual density \$135  
Atari Trackball \$4.95  
Numeric Keypad with Handler \$4.95  
SX212 1200 Baud Modem \$29.95  
XDM121 Daisy Wheel Printer \$99.95  
90 day warranty



### New Item!! 800 Upgrades!!

800 COMPUTER with 288K  
installed only \$175.00

Do-It-Yourself with the JD 288K

Upgrade Kit \$120.00

Includes: 256k PCB  
(2) 16K PCB's (1 cut)  
10K OS Rom (Modified)  
800 Motherboard (Modified)  
You supply the case, keyboard,  
CPU and Power Board.  
Option: \$100 w/o 16K PCB'S

### 800/XL/XE CARTRIDGES \$9.95

ACE OF ACES \*  
ARCHON  
BALLBLAZER  
BATTLEZONE  
BLUE MAX  
CAVERNS OF MARS  
DARK CHAMBERS \*  
DIG DUG  
DESERT FALCON  
DAVIDS MIDNIGHT MAGIC \*  
DONKEY KONG  
DONKEY KONG JR  
EASTERN FRONT '41  
FINAL LEGACY  
FOOD FIGHT \*  
FOOTBALL  
GALAXIAN



GATO  
JOUST  
JUNGLE HUNT  
LODE RUNNER  
MILLPEDE  
MISSILE COMMAND  
MOON PATROL  
NECROMANCER  
ONE ON ONE \*  
FENGO  
RESCUE ON FRONTALUS  
ROBOTRON  
SKY WRITER  
SPACE INVADERS  
STAR RAIDERS II  
TENNIS  
\* = XL/XE ONLY

### ATARI LYNX \$79.95



#### LYNX GAMES SPECIALS!!

BLOCKOUT ..... 24.95  
ELECTROCOP ..... 24.95  
GAUNTLET ..... 24.95  
KLAX ..... 24.95  
ROBOSQUASH ..... 24.95  
ROBOTRON ..... 24.95  
SHANGHAI ..... 29.95  
CHIPS CHALLENGE ..... 29.95  
ISHIDO..... 29.95  
PACK LAND ..... 29.95

**FANNY PACK \$24.95**  
**LARGE CARRY CASE \$23.95**  
**CARRYING POUCH \$19.95**

#### NEW GAME TITLES

STEEL TALONS ..... 39.95  
NFL FOOTBALL ..... 39.95  
BASEBALL HEROES ..... 34.95  
SWITCHBLADE II ..... 39.95  
DIRTY LARRY ..... 39.95  
DRACULA ..... 49.95  
JOUST ..... 39.95  
WORLD CLASS SOCCER... 29.95

We carry a full assortment of games and accessories for the Atari Lynx!

### ST MONITORS

B&C SC1224 \$200  
ATARI SC1224 \$250  
ATARI SC1435 \$365  
ATARI SM147 \$199  
ATARI SM124 \$175

### JRI SIMM PCB

For the 520/1040  
RAM+ 1-4 meg \$99.95

STE, MEGA, OR TT030  
Diagnostic Cart \$75  
Service Manual \$50  
(Price Each - Specify)

### ACCESSORIES for your ST

**Word Flair ST**  
Word Processor  
\$29.95

**FastTalk Modem**  
1200 baud \$29.95  
**Good Starter Modem!**

**MULTI-MEDIA DISK BOX**  
Holds combinations of 3.5" and  
5.25 disks, CD Roms, Tapes and  
Syquest Carts ... \$7.95

**Supercharger IBM**  
Emulator \$425

### PORTFOLIO

hand held computer  
**\$250 with 64K Card!**

#### BSE UNIVERSAL INTERFACE

512K RAM, Parallel/ Serial Ports, and  
Utility ROM \$250.00  
Parallel/Serial Ports & Rom \$99.95  
20 meg FlashDrive \$299.00

PARALLEL INTERFACE .... 49.95  
SERIAL INTERFACE ..... 79.95  
64K MEMORY CARD ..... 99.95  
128K MEMORY CARD ..... 169.95  
AC POWER ADAPTER ..... 9.95  
PC CARD DRIVE ..... 99.95  
PORTFOLIO CHESS ..... 44.95  
INSTANT SPELLER ..... 35.95  
LEATHER CASE ..... 49.95  
VINYL CARRYING CASE ... 24.95  
PORTFOLIO ONLY ..... 219.00

### SPECIAL VALUES FOR YOUR 800/XL/XE

ATARILAB STARTER KIT and ATARILAB  
LIGHT MODULE both for \$14.95  
LOGO Manual set and LOGO cartridge \$9.95  
Visicalc Spreadsheet \$29.95  
AtariWriter & Printer driver both for only \$29.95  
Atariwriter + with spellchecker \$29.95  
ATARI LIGHTGUN with Bug Hunt and  
Barnyard Blaster \$44.95  
410 Program recorder/ States & Capitals \$14.95  
Pilot Language now only \$14.95  
ATARI LightPen \$14.95

### New Low Prices! BULK DISKETTES

10 per package  
Some Contain Old Software  
3-1/2" ..... \$5.00  
\* 5-1/4" (for 810/1050):  
Qty 10 ..... \$2.95  
Qty 100 ..... \$20.00  
Qty 1000 ..... \$100  
\* May Require Disk Notcher

### 800/XL/XE/ST JOYSTICKS

Wico Black Max \$4.95  
ProStick II \$6.95  
Standard Atari \$6.00  
High Score \$9.95  
EPYX 500J \$14.95  
Atari PROLINE \$9.95  
Power Player \$14.95  
Atari Trakball \$7.95  
Paddles \$4.95

**SHIPPING INFORMATION** - Prices do not include shipping and handling. Add \$6.00 for small items (\$10.00 Min. for Canada). Add \$10.00 for disk drive. Calif. res. include 8.25% sales tax. Mastercard and Visa accepted if your telephone is listed in your local phone directory. Orders may be pre-paid with money order, cashier check, or personal check. Personal checks are held for three weeks. International and APO orders must be pre-paid with cashier check or money order. \$20.00 minimum on all orders.

To receive our winter catalogue, please send \$1.00 in coin or stamp to cover the cost of mailing or get it FREE with your order.  
**PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE - NO REFUNDS - ALL SALES ARE FINAL**



# ADVANCED C PROGRAMMING ON THE ATARI 8-BIT, PART 2

MARK MILLER, AC CONTRIBUTING AUTHOR

*AC thanks the Front Range Atari User's Group in Fort Collins, CO for permission to use this article.*

[In Part 1 of this article (AC, Feb. 1993), Mr. Miller briefly surveyed the various C compilers available for the Classic Ataris and outlined the advantages of CC8. This time he wraps up his discourse on CC8 by presenting some of its limitations and giving us some programming examples.]

## Limitations of CC8

Here's my "gripe list" of things in standard C that aren't supported or are limited in CC8:

- The float or double data types, but limited floating point is allowed if Ace C libraries are used. (The float data type should be familiar to experienced programmers, but it has other names in other languages. In Atari BASIC, all numeric variables are floating point variables that can hold real number values. This is what the float type is also. The double type is a double-precision type. You can use floating point numbers if you use the Ace C libraries, but variables used to input your real numbers can't be of type float. The Ace C documents discuss necessary conditions for using floating point values.)
- No bit fields.
- Local variables can't be initialized, as in: `char q[10]="hello";` (inside of a function or compound statement).
- Structs and unions, and their fields, are part of the global name-space.
- Variables can be declared more than once at the same scope level (Ex.: If `int a; int a;` were declarations at the global level, CC8 will declare 'a' twice, which will confuse the linker.).
- Maximum array bound is 6 dimensions (7 dimensions for char and int types). This is still pretty large though.
- Unsigned, long, short, etc. variables can't be declared.
- You can't rename a type using `typedef`.

Despite these limitations, I'm still very impressed with all the features CC8 offers. The compiler is relatively compact- only about 25K-, but the documentation I got didn't state a minimum RAM requirement.

## Runtime C Files: Faster Than Assembly?

Using the Ace C linker, it's possible to create an independent executable file that doesn't require any other file. The way you do this is to link the runtime engine with the rest of the program. This linker also allows you to incorporate assembly code in your C programs. You can either link in absolute (non-relocatable) object files, or put the assembly language right in your C source code. According to both the CC8 and Ace C documentation, the

way you insert assembly into the C source code depends entirely upon the linker you use. Ace C's linker does have a certain '#asm' protocol that's described in the Ace C docs.

Unfortunately, even though your program is compiled into machine language, this doesn't mean you'll get assembly language performance from it. It's been my experience that C programs compiled on an Atari 8-bit are noticeably faster than the equivalent program in BASIC, but there's just no comparison in speed between assembly language and C. Assembly language wins the contest every time. I believe this is because of a general fact of compiled languages. Any time you have to compile a program, the compiled code will generally be less efficient than if the same program were written directly in assembly language by a person. This fact is becoming less true with advances in algorithms for compilers. Most compilers for 16- and 32-bit systems come with optimizers that make the object code nearly as efficient as though the program were written in assembly. Ace C comes with an optimizer, but it mainly helps in making the final object code smaller. In my experience, the Ace C's optimizer doesn't notably enhance execution speed.

## DOS Compatability

If you plan to use CC8, there are some configuration matters you should consider. You don't have to configure the software in any way (like set any settings). What I'm talking about is what DOS you want to run it under and where to put which files. I have a 130XE with an unmodified 1050 drive. I use MyDOS 4.5 and a RAMdisk. The reason I use MyDOS is I like the ability to use sub-directories to organize my files- and there are a lot of files! It turns out MyDOS also comes with a utility that copies whatever files you have in a directory called RAMDISK into the RAMdisk upon bootup. I have it copy the text editor (I use TextPro), compiler, and linker into RAMdisk. This makes loading these large programs *very* fast. If your machine has extended RAM capacity, you should seriously consider doing it this way. (See Mike Jewison's "Fitting Room" column and Jeff McWilliams' "Moonlight Workshop" column elsewhere in this issue for more info on extended RAM. - Ed.)

## Frustrations and Dreams

There are three functions I use heavily when I write C programs with the compilers I use on the minicomputers at school, and which aren't included in the Ace C libraries: `scanf()`, `malloc()`, and `free()`. I've found a substitute for `scanf()` by using other input functions in the libraries. However, what makes a language like C really powerful is the ability to allocate blocks of memory dynamically, so data structures like stacks and binary trees can be created. I've dreamed of running on my 8-bit the same powerful C programs I run on the minicomputers, without

making adaptations to the source code. In order to accomplish my 8-bit programming ambitions in C, I need the malloc() and free() functions. I've found that I'm not able to just copy a C program over from a minicomputer to my 8-bit and compile it. I always have to make some changes to the source code to make it work on different compilers.

When I refer to C's ability to allocate blocks of memory dynamically, what I mean is a different way of storing data than the way most programming languages on the 8-bit do. In Atari BASIC, for example, when you want to use an array, you have to DIMension it first. From then on, you're limited to a certain size constraint, and you can only have one data type in it: characters for a string array, or numbers for a numeric array. A typical thing I do is define a struct (remember, a struct is a collection of variables that can be of different data types), and then call the malloc() function to allocate space for the struct *somewhere* in memory. Memory that's dynamically allocated isn't necessarily sequential (even though that's how it usually turns out on personal computers). A block of memory somewhere is found to be open, and a pointer (address) to that block is returned, so data can be put in that block (values for the fields of the struct, for example) through indirection.

### Linked-List Via Dynamic Allocation

Let's use a simple data structure as an example: a linked-list. This data structure is formed by dynamically allocating blocks of memory for structs one by one. Each struct contains a pointer to the struct type containing it. For this example I'll call this pointer field "next". The construction of a linked list goes thus: Declare a structure type, I'll name it "node", a couple of local pointers, head, and a temporary pointer, I'll call it p, of type "pointer to struct node." The pointers "head" and "p" are declared outside the struct. Don't confuse these with being fields in struct node. A pointer called "head" (or some other name) is usually declared so it can be used as a beginning reference point for the list. The pointer "p" is used to create the linked-list by gradually travelling down the list, and being used as a reference point for more blocks to be added to the list. To start the construction of the list, you allocate a block of memory the size of struct node using malloc(). Malloc() will return an address to a free space it has found. Assign this address to the pointer "head" and "p". Then allocate another block, using malloc(), and assign the returned address to the "next" field in the block pointed to by "p".

This procedure has thus created a linked-list, since the two blocks have been connected by a pointer. If you want to add more blocks to the list, move "p" to the next block by saying "p = p->next". Fields in an allocated block can be referenced through the representation of the declared struct. What the ">" does here is say "Go to the block pointed to by 'p', and go to the 'next' field in that block." Since "next" contains the address of the next block allocated, it makes "p" point to the next allocated space. Do *not* also move "head". Its purpose is to be a reference to the beginning of this list, so that if you ever want to travel down the list again from the beginning, you can by saying "p = head". After "p" has been moved to this next allocated block, you proceed:

```
/* declare struct node */
struct node
$(
    char ch;
    int num;
    char name[15];
    struct node *next; /* this is a pointer to this structure, of type */
                        /* "struct node" */
$)

Now say we do the following in main():
main()
$(
    /* declare pointers to structure of type "struct node" */
    struct node *head, *p;
    /* go out and get some memory that is just the right size for a */
    /* struct of type node */
    head = (struct node *) malloc(sizeof(struct node));
    p = head;
    /* do it again and assign the returned address to the pointer */
    /* in the allocated block */
    p->next = (struct node *) malloc(sizeof(struct node));
    /* move temporary pointer to newly allocated block */
    p = p->next;
    /* allocate a 3rd block and assign the returned address to the */
    /* pointer in the block */
    p->next = (struct node *) malloc(sizeof(struct node));
    /* The operations of moving the pointer p and allocating a new */
    /* block and assigning the address to the "next" field is usually */
    /* done in a loop */
$)
```

The "(struct node \*)" you see after the "=" is a type cast. What this does is insure that the value on the right side of the "=" is of the correct data type for the variable on the left side of the "=". The "sizeof(struct node)" function call returns the total number of bytes all the variables in struct node require.

Since a linked-list can conceptually have any number of memory blocks linked together, how do you determine how long it is? The usual method is to assign the pointer field ("next" in this case) of the last block in the list to a value called NULL. This NULL value is a specific value, usually 0. This is used so that when a function is searching down the list, it can know when the list ends by checking if the temporary pointer "p" has become NULL, or if NULL is what the "next" pointer points to. Some examples of searching a linked-list and testing for NULL may look like this: "while(p != NULL) ..." or "while(p->next != NULL) ..."

### Why A Linked-List?

Besides allocating new blocks and linking them together, I can put values in these blocks, in the other fields. So I could do expressions like: p->ch = 'a'; strcpy(p->next->name, "me"); p->num = 16;

The power of linked-lists is you don't have to dimension *anything*! In principle you can get as many memory blocks as you want. All you have to do is ask for them by calling malloc(). Notice I said this is the "principle." With the functions I've written, there's a limited amount of space for this type of allocated memory, since I have them use Page 6, which only has 256 bytes available. Presently, these functions are good for experimenting with this concept of dynamically allocating memory for different uses. On the other hand, as far as the amount of space a struct takes up, 256 bytes is nothing to sneeze at either. I've run C programs on my 8-bit, using these functions, that have implemented linked-lists (one of my programs did two at once), and binary trees. And I haven't run out of space yet. You just won't be able to do anything real large with these functions I've written.

My first attempt at writing the malloc() and free() functions worked...but it was quick and dirty. For malloc(), I used a static variable that kept track of the next free location in memory. When I wanted to de-allocate something, I passed in the pointer to the block I wanted to de-allocate, and the size of it, to free(), which cleared the block to zeros. But the static variable would keep the same value, and just keep increasing (going forward through memory) each time I used malloc(). I could see that for the long haul, this just wouldn't do. My real goal was to create a set of functions that would support dynamic allocation. In other words, if there was a free spot somewhere in Page 6 of the right size, that spot would be a likely candidate to put something in. So if I freed a block, and later wanted to allocate another block that was the same size or smaller, it would return a pointer to a block that was previously freed. This would make the maximum use of the 256 bytes available.

## Dynamic Allocation: A Breakthrough

A friend of mine gave me an idea one day when I said I didn't want to keep passing in the size of the data to de-allocate, to free(). He suggested I put a byte before each allocated block, representing the length. That way, free() would be able to determine the size itself. I expanded on this idea to create more advanced versions of malloc() and free() that do the job of dynamic memory allocation within the boundaries of Page 6 and are still fairly memory-efficient. The concept of using a byte to store the size of an allocated area was the missing piece I needed.

Now I'll show you the source code of the set of functions I wrote enabling CC8 to dynamically allocate memory. Further, I'll demonstrate that it's possible to have almost identical source code of a program compiled on two different systems: a minicomputer and an Atari 8-bit. Following are two samples of the same program (written by my friend Darryl May). One compiles and runs on a minicomputer here at Colorado State University. The other is a slightly modified version that compiles and runs on an Atari 8-bit using CC8 with the Ace C linker and libraries. The CC8 version uses the following set of functions (which I wrote).

```
/* ALMLIB.C for CC8, by Mark Miller 5/27/91. */
#define NULL 0
#define MEMSTART 1536
#define MEMSIZE 256
/* num of bytes for alloc info. */
#define INFOLEN 2
#define FREE_SPC 1
#define USED_SPC 2
/* Does housekeeping on memory space. If there are
consecutive memory spaces that are freed, it will
consolidate these spaces into one free space.
*/
update()
$(
/* ptr to alloc info.; ptr to first free space found */
char *loc, *tempaddr;
/* size of a space; accum for size of consecutive
spaces */
int size, templen;
/* flag for first non-free space found (after free
space(s) found) */
int done;

loc = MEMSTART;
templen = 0;
/* search through memory space until end, for free
space to consolidate */
```

```
while(loc < MEMSTART+MEMSIZE)
$(
if(*loc == FREE_SPC)
$(
/* if space is free, search for consecutive free
spaces */
size = *(loc+1);
/* hold first free-space pos */
tempaddr = loc;
/* get size of this space */
templen = size;
/* get to next info. bytes */
loc += size+INFOLEN;
done = 0;
while((loc < MEMSTART+MEMSIZE) && !done)
$(
/* if next space is free, add its free space to
accum */
if(*loc == FREE_SPC)
$(
size = *(loc+1);
templen += size+INFOLEN;
$) /* if */
else
/* otherwise, done with present search */
done = 1;
/* go to next info. bytes */
loc += size+INFOLEN;
$) /* while */
/* if more than one free space was found, record
this free space at first free-space pos */
if(templen != size)
*(tempaddr+1) = templen;
templen = 0;
$) /* if */
else
/* if space is not free, go to next info. bytes
(next space) */
loc += size+INFOLEN;
$) /* while */
$) /* update */
/* Finds a free space for something of specified size
(data type does not matter). If no free space can be
found with specified size, malloc will return NULL.
Currently malloc only works in the 256 byte space of
Page 6 (location 1536 decimal).
*/
char *malloc(size)
int size;
$(
/* ptr to info. bytes; ptr to allocated space */
char *loc, *bloc;
/* amt of space free/used */
int space;
/* flag for when free space found and allocated; flag
for when update() is called */
int done, updated;
loc = MEMSTART;
/* if memory area is completely unused, put in info.
bytes, set to free space */
if(*loc == 0)
$(
*loc = FREE_SPC;
*(loc+1) = MEMSIZE-2;
$) /* if */
bloc = NULL;
updated = 0;
done = 0;
while(!done)
$(
/* search for free space of specified size (or more) */
while((loc < MEMSTART+MEMSIZE) && !done)
$(
space = *(loc+1);
/* if space is free, allocate it and return ptr to
```



```

        it. Also put modified info. bytes after allocated
        space */
if((*loc == FREE_SPC) && ((space == size) ||
    (space >= size+INFOLEN)))
$(
    *loc = USED_SPC;
    *(loc+1) = size;
    if(space >= size+INFOLEN)
    $(
        *(loc+INFOLEN+size) = FREE_SPC;
        *(loc+INFOLEN+1+size) = space-size-INFOLEN;
    $) /* if */
    bloc = loc+INFOLEN;
    done = 1;
$) /* if */
else
    /* otherwise, go to next info. bytes */
    loc += space+INFOLEN;
$) /* while */
if(!updated && !done)
$(
    update();
    updated = 1;
    loc = MEMSTART;
$)
else
    done = 1;
$) /* while */
return bloc;
$) /* malloc */
/* Deallocates space pointed to by addr. */
free(addr)
char *addr;
$(
    int size;
    if((addr >= MEMSTART) && (addr < MEMSTART+MEMSIZE))
        *(addr-2) = FREE_SPC;
$) /* free */
/* Allocation Memory Reset:
   Sets all bytes in memory space to 0, just in case any
   data is there.
*/
almreset()
$(
    clear(MEMSTART-1, MEMSIZE+1);
$)
/* Allocation Memory Dump:
   This function is provided for program debugging
   purposes. It displays, in decimal form, what is in the
   memory area used for dynamic memory allocation by
   malloc().
*/
almdump()
$(
    char *addr;
    printf("Allocation memory dump from location\n");
    printf("%d to %d, in bytes:\n", MEMSTART, MEMSTART +
        MEMSIZE);
    for(addr = MEMSTART; addr <= MEMSTART+MEMSIZE; addr++)
        printf("%d ", *addr);
    printf("\nPress RETURN to continue");
    getchar();
$) /* almdump */

```

If you'd like to use the malloc(), free(), almreset(), and almdump() functions [update() is used by malloc(); you shouldn't need to use it yourself], just compile the above source code. I've named this library PTRLIB.C on my disk (when compiled it's filename is PTRLIB.CCC). When your program needs these functions, link PTRLIB.CCC with your compiled program, using a linker. CC8 doesn't mind if you call functions that don't exist in your source code. The linker deals with that.

Notice the part in malloc() that says when you want to

allocate space for something, the data type doesn't matter, except when it comes to determining the size of the block you want to allocate. You could write something like the following, and it would work like a regular character array:

```

main()
$(
    char *ch;
    ch = (char *) malloc(5*sizeof(char));
$)

```

This would do essentially the same thing as declaring "char ch[5];" except you would have to remember to free(ch) when you were done with it (remember this allocated space is in Page 6). This is just to illustrate that you don't have to use just structs with malloc(). All malloc() cares about is how many bytes you want.

## Pascal's Triangle: A C Comparison

Now for the example program. This program by Darryl May (thanks, Darryl!) displays Pascal's triangle- which comes from the Binomial Theorem- for however many levels you want. It uses linked-lists to do this. First, here's the version that runs on a minicomputer here at C.S.U.:

```

/* CS253
   Assignment 5 - c
   By Darryl May
   April 25th 1990
   */
#include <stdio.h>
struct lnode
{
    int value;
    struct lnode *next;
};
struct lnode *new_row;
struct lnode *old_row;
struct lnode *print_row;
struct lnode *head_old;
struct lnode *head_new;
struct lnode *temp;
main(argc, argv)
    int argc;
    char *argv[];
{
    int number_of_rows;
    unsigned int v1;
    if (argc == 2) number_of_rows = atoi(argv[1]);
    else scanf("%d", &number_of_rows);
    /* Put a value of 1 into old row to get things started */
    old_row = (struct lnode *) malloc(sizeof(struct lnode));
    old_row->value = 1;
    old_row->next = NULL;
    head_new = old_row;
    printf("%u\n", old_row->value);
    while(number_of_rows--) /* Main loop */
    {
        /* Start the new row off with a value of 1 */
        new_row = (struct lnode *) malloc(sizeof(struct lnode));
        print_row = new_row;
        head_old = old_row;
        head_new = new_row;
        new_row->value = 1;
        while(old_row->next) /* Keep building new row until */
        {
            /* all the values in old row */
            v1 = old_row->value; /* have been used. */
            old_row = old_row->next;
            new_row->next = (struct lnode *) malloc(sizeof(struct lnode));
            new_row = new_row->next;
            new_row->value = v1 + old_row->value;
        }
        /* End the new row with a value of 1. */
    }
}

```

```

    new_row->next=(struct lnode *) malloc(sizeof(struct
        lnode));
    new_row = new_row -> next;
    new_row -> value = 1;
    new_row -> next = NULL;
/* Print out the new row. */
    printf("%u ",print_row -> value);
    while(print_row -> next)
    {
        print_row = print_row -> next;
        printf("%u ",print_row -> value);
    }
    printf("\n");
/* Change the old row pointer to point to the head of the
    new row. */
    old_row = head_new;
/* Return the memory used by the new row back to the
    system. */
    temp = head_old -> next;
    free(head_old);
    while(temp)
    {
        head_old = temp;
        temp = head_old -> next;
        free(head_old);
    }
    old_row = head_new;
} /* End of the main loop */
/* Return the memory used by the old row back to the
    system. */
temp = head_new -> next;
free(head_new);
while(temp)
{
    head_new = temp;
    temp = head_new -> next;
    free(head_new);
}
} /* End of program. */

```

Now here's the Atari 8-bit version, slightly modified so it'll compile with CC8:

/\* Atari 8-bit version, using CC8 with Ace C linker and libraries.

```

CS253
Assignment 5 - c
By Darryl May
April 25th 1990
*/
#define NULL 0
struct lnode
$(
    int value;
    struct lnode *next;
$);
struct lnode *new_row;
struct lnode *old_row;
struct lnode *print_row;
struct lnode *head_old;
struct lnode *head_new;
struct lnode *temp;
main()
$(
    int number_of_rows;
    int v1;
    char numinp[3];

```

```

    printf("Enter number of rows to do\n");
    gets(numinp);
    number_of_rows = val(numinp);
/* Put a value of 1 into old row to get things started */
    old_row=(struct lnode *) malloc(sizeof(struct lnode));
    old_row -> value = 1;
    old_row -> next = NULL;
    head_new = old_row;
    printf("%d\n",old_row -> value);
    while(number_of_rows--) /* Main loop */
    $(
/* Start the new row off with a value of 1 */
        new_row=(struct lnode*) malloc(sizeof(struct lnode));
        print_row = new_row;
        head_old = old_row;
        head_new = new_row;
        new_row -> value = 1;
        while(old_row -> next) /* Keep building new row until */
        $(
            /* all the values in old row */
            v1 = old_row -> value; /* have been used. */
            old_row = old_row -> next;
            new_row->next=(struct lnode *) malloc(sizeof(struct
                lnode));
            new_row = new_row -> next;
            new_row -> value = v1 + old_row -> value;
        $)
/* End the new row with a value of 1. */
        new_row->next = (struct lnode *) malloc(sizeof(struct
            lnode));
        new_row = new_row -> next;
        new_row -> value = 1;
        new_row -> next = NULL;
/* Print out the new row. */
        printf("%d ",print_row -> value);
        while(print_row -> next)
        $(
            print_row = print_row -> next;
            printf("%d ",print_row -> value);
        $)
        printf("\n");
/* Change the old row pointer to point to the head of the
        new row. */
        old_row = head_new;
/* Return the memory used by the new row back to the
        system. */
        temp = head_old -> next;
        free(head_old);
        while(temp)
        $(
            head_old = temp;
            temp = head_old -> next;
            free(head_old);
        $)
        old_row = head_new;
    $) /* End of the main loop */
/* Return the memory used by the old row back to the
    system. */
    temp = head_new -> next;
    free(head_new);
    while(temp)
    $(
        head_new = temp;
        temp = head_new -> next;
        free(head_new);
    $)
    getchar();
$) /* End of program. */

```



# A Look At SYNCHROMESH

GARY MATTESON (aka "THE WAGON MASTER") AC Staff Reviewer

## Atari In The Wild West

My connections with the Atari began in 1982 when "2001", "An Electronic Odyssey of Games, Gadgets & Gizmos" opened their doors here in Norfolk, Nebraska. Dick and Debbie Day, the then proprietors of said store recently brought to my attention how, out of all the people who visited their new store during the grand opening, I was first to buy an Atari computer—a 400 with a 32k Axlon expansion board that yielded a whopping 48K of memory, plus a 410 cassette drive. Like many of my fellow Atarians at the time, my billfold was not overloaded with greenbacks. Obviously not a pleasing situation, but ultimately it worked to my advantage because I was forced to become a careful buyer.

Those were the days before any of us heard slogans like "Power Without the Price": they definitely were days of getting better educated and in more ways than I could have imagined. Few references were available then, although the manuals that accompanied this first generation of Classic Ataris were considerably more user-friendly and helpful than the manuals with the ST of later years. So, like many others bedazzled by the possibilities of the little 8-bit wonder, I read everything I could get in my "shaking with excitement hands." Pre-*ANTIC* days, pre-*Compute!* days, pre-everything days: still have your copies of *Santa Cruz Educational Software*? Remember the "Tricky Tutorials" on tape? Hey—I've got mine! But tape has its limitations, and out here in the hot Nebraska sun I thirsted for a disk drive.

## Corraling The Indus

The day arrived when I could no longer put off buying my first disk drive. Initially I considered an Atari, but with over (gasp!) one year of experience and an equal amount of time browsing articles and ads, I purchased an Indus GT. Perhaps a year later another GT joined the first one in my "komputer kornet", as I referred to it then. With the arrival of this second drive there came a surprise: a new version of OSS's DOS-XL: rev. 2.35I2, and something else called "Synchronmesh".

Indus Systems of Chatsworth, CA had sent me a product I had never heard of, but I soon realized and then appreciated its value: SPEED! Basically, Synchronmesh and DOS-XL (in the manufacturers own words) are "...two completely independent software packages which require each other in order to function to their fullest extent." What the two accomplished together was "to increase the transfer of data" between my Atari computer and the Indus GT "at speeds, multiple times faster than ordinary disk drive data transfers."

## Using SynchroMesh

You begin by install this file called INITSYNC.COM onto your boot disk. Upon bootup, that file would initialize designated programs on any disk with the file SYNCBOOT.SYS; the effect was similar to what would happen to cattle if I cracked a whip over their heads and hollered "Git along little doggies!", only faster. Once Synchronmesh has loaded, everything after that loads lightning quick. Recall for a moment the

beep sounds you normally hear when a program or data is loading. With Synchronmesh, the "beeps" are transformed into sounds...well, a little like a telegrapher's key at work.

Synchronmesh is "density smart", and you can turn it "on" or "off". Other features: no difference in disk formatting, booting or copying procedures; those with a RAM Charger experience an even greater increase in performance. Synchronmesh is useable with the entire Atari line, the 400/800/XL/XE, but you *must* have an Indus GT drive. No other brand of disk drive will work with Synchronmesh. Gosh, imagine that, its only disadvantage. DOS-XL will work on other brands of drives, but without the hi-speed capability offered by Synchronmesh on the Indus GT.

Another Indus product that I missed out on: a printed circuit board that piggy-backed onto the one inside the GT drive, allowing the Atari to use CP/M. Switching between CP/M and DOS was supposed to be accomplished from the keyboard. If there are still some Indus users out there, I'd be delighted if anyone who used this product could write to *AC* about their experience. How well did it function? Did it provide access to useful programs, meeting a need not covered by standard Atari programs? Is it still available? Write to me c/o \*\*AC\*\*, we'll see if there are still a few diehard Indus GT users out there.

Happy Trails to you...

*Editor's Note: Indus Systems—famous for its "Turn Your Atari Into A Ferrari" ads in ANTIC and Analog, probably chose the Synchronmesh name to fit its racy marketing image. By 1987 the company had burned itself out in competition with the Atari 1050. Synchronmesh is basically a sector skewing scheme similar to the many upgrades available for 1050 drives (ICD's U/S Doubler, CSS's Super Archiver, Happy 1050, etc.). Similar speed performance could be obtained at a much lower price by purchasing a 1050 and the desired upgrade separately.*

*The interesting thing about Synchronmesh is that the speed enhancement was accomplished via software, while corresponding upgrades to the 1050 were primarily hardware. Unfortunately, the Indus software upgrade required perfect communications between the computer and drive, while the 1050 hardware upgrades are more robust. According to AC's Technical Consultant, Bob Puff, any minor glitch on the SIO could cause the Indus drive to lock up when using Synchronmesh. There were also vague reports of compatability problems between Synchronmesh and DOS's from other manufacturers.*

*In those early days of Atari's catapult to 8-bit success, there were a dozen ways to solve every problem and no one knew which way would be best. Manufacturers sprang into the market with products reflecting their particular philosophy. The fact that there's a lively market for used 1050's today, and that hardware upgrades for the 1050 are still readily available, suggests the hardware approach to speed enhancement was the final winner over the software method represented by Indus. Nevertheless the sleek and sporty Indus still has its coterie of followers, and you'll see the occasional GT turning up at swap meets. -BP*





# News Roundup: Happenings In The 8-Bit World

BEN POEHLAND MANAGING EDITOR

## CompuServe Lowers Rates

According to Ron Luks, Atari Forum Manager on CompuServe, beginning March 1993 CIS offers its subscribers a Standard Pricing Plan under which members pay a flat monthly fee of \$8.95 for unlimited connect-time use of 36 basic services such as travel, shopping, investment, and games. When using CompuServe's other services, members will pay an hourly charge of \$8 for 1200 or 2400 baud access, and \$16 for 9600 baud access. Previously, members paid a monthly fee of \$7.95 and hourly charges of \$12.80 and \$22.80 for access at 1200/2400 and 9600 baud, respectively. Ron notes that while this new BASIC SERVICES plan isn't mandatory for members, the lower rates certainly make it more advantageous for most users to switch to the new pricing structure. Ron further related that the new fee schedule includes up to 60 3-page e-mails/month for free with no time of day restrictions and a small additional charge for overage. CompuServe users can type GO CHOICES for details on the new fee structure, or contact Ron Luks [76703,254] if you have questions.

AC applauds this effort by CIS to make its fees more competitive with other commercial networks such as the General Electric News and Information Exchange (GENie). GENie also has a flat-rate fee schedule of \$4.95/month with unlimited use of certain services such as e-mail at certain times of the day. However, a variety of restrictions apply to the services of both networks, and we urge users to explore fees in detail and to shop around, as fees are now very competitive but also a bit complicated. CompuServe has long been noted as "the place to be" for Atari 8-bitters. AC's Technical Consultant, Bob Puff (who also runs Computer Software Services) is an Atari Forum sysop on CIS, and Don LeBow—another notable on CIS—serves as one of our Correspondents.

Although GENie has potentially greater resources—especially in the form of its large library of downloadable 8-bit files—the Atari management there has shown no interest in the revival movement represented by this magazine and has generally allowed things in the ATARI8 area to decay. Recently, a group of 8-bit users on GENie attracted the attention of Darlah Potechin with proposals to revive the 8-bit area there and perhaps take on some new sysops. With this latest aggressive fee reduction by CIS, we at AC are observing the proceedings on both networks with keen interest.

## Good-Bye, Innovative Concepts

AC regrets to report several documented cases of ripoffs perpetuated by Innovative Concepts of Warren, Michigan. Part of the complaints we received indicated that proprietor Mark Elliot appeared to be hiding behind an answering machine and not returning calls. Our Managing Editor left a message on the IC answering machine and received similar treatment.

As nearly as we can tell, Innovative Concepts appears to have quietly slipped its moorings around August of 1992—yet another casualty of the dreadful Summer of '92 which snuffed out so many Atari vendors and developers and took a toll on magazines as well. Unfortunately, its proprietor failed to announce that fact to the community and apparently in some instances continued to accept money for services and merchandise that it never delivered. Unless this magazine receives some notice from Innovative Concepts indicating their Atari 8-bit business is once again up and running, we must regrettably advise readers to refrain from placing orders with IC. Really a shame, this company used to sell some really great stuff for the 8-bit.

## TEXTPRO, Where Art Thou?

Afficionados of Ron Ritchie's TextPro wordprocessor have been waiting for some time for him to release the latest version, TextPro 5.0. Word is that beta-testing of 5.0 is complete and the only present obstacle to final distribution is that the manual is being re-written. We're hoping the long wait will be over soon!

## RT-8 Cart Clone Delayed

In the Dec. '92 issue of AC, Brad Koda of Best Electronics announced in an ad a replacement for ICD's discontinued Real-Time8 clock cartridge. There is a pent-up demand for this product, especially among BBS sysops who will practically kill you for one. We had hoped to see this product launched in early 1993, but according to Brad Koda there's a minor snag delaying its introduction. Apparently the firmware for the cart contains a bug that causes the clock to lose time, like several minutes a week. Software development for the new cart is being done in England. In late February Brad travelled to England, where—among other things—we hope he got his timepiece fixed. An RT-8 cart clone is a welcome product; we predict it will sell briskly once it reaches the market.



# Swap 'n' Shop

## A Flea Market for All Atari 8-Bit Users!

**SALE:** hard drives & controller cards. Seagate ST-213 5-megs MFM HH 2-heads, \$50 firm. 10-meg MFM 4-heads \$100 firm. IBM controller card for 10-meg drive, \$50 firm. OMTI MFM-SCSI controller card \$50 firm. Barry Shortridge, P.O. Box 210, Mouth Card KY 41548. 606-835-2100.

**SEEKING ANYONE** with knowledge of the Minitel Terminal Emulator. The Minitel was used in the USA (New York & Chicago). It is a view-data program using a 75/1220-Bd modem & has 80 columns with ASCII mode for BBS. Daniel Carrodano, 202 Impasse des Fougères, La Bouverie, 83520 Roquebrune/Argens FRANCE.

**SALE:** Atari 8-bit & ST software. Most titles in original boxes, some never opened. Assorted loose 8-bit carts also available. For free list with prices send SASE to: Craig McGrath, P.O. Box 673, Forked River NJ 08731.

**THE CLUB II BBS:** central repository of 8-bit software, 100's of files & 100's of megs for more! 313-334-8877, 300-14400-Bd, no quotas! Terry Conklin, 138 Regent, Lansing MI 48912.

**ICD MIO** for sale: 1-meg RAM, pwr supply & cable, \$100 or B/O. Also SpartaDOS Construction Set & SpartaDOS Toolkit, brand new w/manuals, \$20 each or \$35/both. Shane Rolin, 240A Main St., Imperial PA 15125. 412-695-8080.

**WANTED:** ANALOG issue #1 (Jan/Feb. 1981), ANALOG issue #5 (Sep/Oct. 1981). Also seeking ANTIC Vol. 1 issues 2,3,4 (all 1982). Will pay top price for mags in good condition. Ben Poehland 179 Sprout Road/Rt. 352, Frazer PA 19355.

*Ads are FREE to paid subscribers of AC. Please refer to the February 1993 issue for Rules regarding placement of personal ads.*

### ZTM Software Manufacturers

#### P.O. Box 240, Forked River, NJ 07831

Atari XL/XE Software Developers & European Distributors

#### Exciting New Commercial Action Games From Europe

CHANGE	\$15	ROBBO	\$15	ADAX	\$15
JET ACTION	\$15	DAGOBAR	\$15	HANS KLOSS	\$15
GOLD HUNTER	\$15	THE CURSE (GRAPHIC ADVENTURE)			\$15

#### Just Developed by ZTM Software Manufacturers!

#### Atari Tape Backup Program (Disk software) \$40

This newly developed program will allow you to back up any of your precious file structured diskettes onto tape for safe keeping in the event you encounter a disk crash. The program supports the following features:

- Internal labeling of your back-up tape.
- Wildcard specification on backups.
- Unique tape directory listing.
- Program recorder is required.
- Multiple tape support of your disk to tape transfers (no more error 162).
- Standard Atari condition codes.
- Quality cassette tapes are suggested.

#### Product Information Line: (908)-928-9371

We accept money orders, personal checks, and an open account upon approval for faster service of future purchases by calling our product information line. - Shipping: \$3.75 shipping and handling on all purchases. In Canada or P.R. add add \$7.50 for shipping and handling. Foreign shipping extra. N.J. residents include 6% sales tax. All sales final. Defective merchandise will be replaced when returned in its original cartons with dated sales invoice and explanation of defect. Prices subject to change without notice.

**ENERGIZE YOUR 8-BIT: Plug into AC! ATARI CLASSICS, that is! Subscribe NOW !!**

Name: \_\_\_\_\_

Address: \_\_\_\_\_

City: \_\_\_\_\_ State/Country \_\_\_\_\_

ZIP/Postal Code \_\_\_\_\_ Phone: \_\_\_\_\_

E-mail address: \_\_\_\_\_

All subscriptions are for one year. Note: a one-year subscription to AC may contain fewer than 6 issues if circulation fails to reach 500 by 1st qtr. '93. Subscription rates:

<input type="checkbox"/> USA 3rd Class Mail	\$25 for 1 year	<input type="checkbox"/> Europe/Mediterranean AIRMAIL	\$38 for 1 year
<input type="checkbox"/> Canada 3rd Class Mail	\$30 for 1 year	<input type="checkbox"/> Asia/Pacific Area AIRMAIL	\$40 for 1 year
<input type="checkbox"/> Foreign (All) 3rd Class Mail	\$32 for 1 year	<input type="checkbox"/> AC Software Disk (global)	\$ 9 for 1 year

The AC Software disk will be distributed 3 times/year, concurrently with every other issue of the magazine. Disks will be DS/SD flippies and will contain all the type-in software from the current + previous issue plus selections from the Public Domain as space permits.

☐ Personal check    ☐ M.O    ☐ I.M.O    ☐ I.P.M.O.    ☐ VISA    ☐ MasterCard

Card Number \_\_\_\_\_ Exp. Date: \_\_\_\_\_

Your Signature \_\_\_\_\_ Date: \_\_\_\_\_

(Note: credit card orders will be charged a \$2 processing fee on orders under \$100, \$3 on orders over \$100.) Credit card orders may be phoned in directly to Unicorn Publications: 313-973-8825. Please make checks or money orders payable in U.S. Funds. DO NOT SEND CASH. Clip this form and return to:

Atari Classics, 179 Sproul Road/Rt. 352, Frazer, PA 19355-1958USA    ATTN: Circulation Editor

ACV2N2

Atari Classics  
179 Sproul/Rt. 352  
Frazer, PA 19355

**Bulk Rate  
US Postage  
Paid Permit #87  
Ann Arbor, MI**

Address Correction Requested

**Dated material—Please Rush!**

